Combining Machine Learning Defenses without Conflicts

Anonymous authors
Paper under double-blind review

Abstract

Machine learning (ML) models require protection against various risks to security, privacy, and fairness. Real-life ML models need simultaneous protection against multiple risks, necessitating combining multiple defenses *effectively*, without incurring significant drop in the effectiveness of the constituent defenses. We present a systematization of existing work based on *how defenses are combined*, and *how they interact*. We then identify unexplored combinations, and evaluate combination techniques to identify their limitations. Using these insights, we present, Def\Con, a combination technique which is (a) *accurate* (correctly identifies whether a combination is effective or not), (b) *scalable* (allows combining multiple defenses), (c) *non-invasive* (allows combining existing defenses without modification), and (d) *general* (is applicable to different types of defenses). We show that Def\Con achieves 90% accuracy on eight combinations from prior work, and 86% in 30 unexplored combinations which we empirically evaluated.

1 Introduction

Machine learning (ML) models are susceptible to a wide range of risks to security (Papernot et al., 2018; Tian et al., 2022), privacy (De Cristofaro, 2020; Hu et al., 2022), and fairness (Mehrabi et al., 2021; Pessach & Shmueli, 2022). Several defenses have been proposed to mitigate them. Real-life models require simultaneous protection against multiple risks. But defenses designed to protect against one risk (Li et al., 2023a; Machado et al., 2021; De Cristofaro, 2020; Mehrabi et al., 2021) may impact susceptibility to other unrelated risks (Duddu et al., 2024a). This raises the question of how to combine defenses *effectively*, without incurring a significant drop in the protection provided by each constituent defense. Practitioners need effective *combination techniques*, either by *modifying existing defenses* (invasive), or by identifying whether *existing defenses can be combined without modification* (non-invasive). Prior work is either limited to specific defenses (e.g., Szyller & Asokan (2023); Chen et al. (2023); Fioretto et al. (2022); Noppel & Wressnegger (2024)), or study interactions among defenses with risks (Duddu et al., 2024a; Gittens et al., 2022). No prior work provides a way to quickly determine if *defenses can be combined effectively*.

We first systematically survey existing work on combining defenses based on: (a) *how defenses are combined* (i.e., what combination technique was used), and (b) *how they interact* (i.e., whether they conflict or align). We then identify previously unexplored combinations, and evaluate prior combination techniques to identify their limitations.

Non-invasive combination techniques are easier to deploy as they do not require expert knowledge from practitioners. Therefore, we identify "*mutually exclusive placement*" (Yaghini et al., 2023) as a promising technique. It presumes that two defenses can be effectively combined iff they *operate on different stages in the ML pipeline*: pre-, in-, and post-training. However, it can result in ineffective combinations: (a) later-stage defenses can conflict with earlier ones (e.g., model or dataset watermarking with adversarial training or differential privacy (Szyller & Asokan, 2023)), and (b) same-stage defenses may still be compatible (see §7).

Based on these insights, we present Def\Con, a technique which is (i) *accurate* (correctly identifies whether a combination is effective or not), (ii) *scalable* (allows two or more defenses to be combined), (iii) *non-invasive* (does not require changes to the defenses being combined), and (iv) *general* (applicable to different types of defenses). Def\Con is inspired by mutually exclusive placement (aka naïve technique), but overcomes its

limitations by explicitly addressing the reasons that underlie conflicts among defenses: a later-stage defense either (a) mitigates a risk re-purposed as a defense by an early-stage defense, or (b) overrides changes made by an early-stage defense (Szyller & Asokan, 2023). We claim the following contributions:

1. a *systematization* of prior work based on combination techniques, and types of resulting interactions; (§4)
2. *identifying* unexplored combinations, and *evaluating* prior techniques for limitations.; (§5.1 and §5.2)
3. DEF\CON[1], a scalable, non-invasive, and general combination technique (§5.3), which is more accurate than the naïve technique, with a balanced accuracy of
   - 90% (DEF\CON) vs. 40% (naïve) using eight combinations from prior work as ground truth, (§7.2),
   - 86% (DEF\CON) vs. 36% (naïve) via empirical evaluation of 30 unexplored combinations (§7.3 and 7.4).

   DEF\CON constitutes an inexpensive and fast technique for practitioners to determine if a combination of existing defenses without modification can be effective, without using expensive empirical evaluation.

2 Background: ML Notations

Consider X as the space of all possible input data records (e.g., images, text prompts) and Y as the space of corresponding outputs (e.g., classification labels for classifiers, predicted next tokens for generative models). An ML model is a function f^θ which maps x to y, i.e., $f^\theta : X \rightarrow Y$ where θ indicates the model's parameters. Hereafter, we denote f^θ by simply writing f. We consider two datasets of the form $(x,\ y)$, where x is the input data record and y is the output, for training an ML model using training dataset ($\mathcal{D}_{tr}$) and evaluate the model on test dataset ($\mathcal{D}_{te}$). We focus our evaluation (§7) on classifier models, and describe the training and inference for classifiers.

Training. We iteratively update θ using $(x, y) \in \mathcal{D}_{tr}$ over multiple epochs to minimize some objective function $\mathcal{C}$: $\min_\theta\ l(f(x), y; \theta) + \lambda\, R(\theta)$ where $l(f(x), y)$ is the prediction error on x for the ground truth y. $R(\theta)$ is the regularization function which restricts θ from taking large values and λ is a hyperparameter to control regularization. The parameters are updated as: $\theta := \theta - \alpha \frac{\partial \mathcal{C}}{\partial \theta}$ where α is the learning rate.

Inference. We measure the utility of f using its accuracy on $\mathcal{D}_{te}$ computed as

$$\phi_u(f, \mathcal{D}_{\text{test}}) = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{(x,y)\in\mathcal{D}_{\text{test}}} \mathbb{I}\left\{\hat{f}(x) = y\right\}$$

where $\hat{f}(x)$ is the most likely class. If ϕ_{u} is acceptable, f is deployed to provide predictions for input x, represented by $f(x)$ for the probability vector across different classes.

3 Framework for Systematization

Given a list of defenses whose combinations are explored (§3.1), we present a framework to systematize prior work based on *how the defenses are combined* (§3.2), and *how they interact* (§3.3). We then discuss the completeness of our framework and how to extend it (§3.4).

3.1 Defenses being Combined

We describe various defenses proposed to mitigate risks to ML models in the presence of an adversary ($\mathcal{A}dv$). As part of our systematization (§4), we will later enumerate all possible pairwise combination of these defenses.

Evasion robustness (EvsnRob) protects against *evasion*, which forces f to misclassify an input x by adding perturbation δ_{rob} (aka adversarial examples) (Machado et al., 2021; Madry et al., 2018). Here, $\delta_{rob} = argmax_{\delta_{rob}} l(f(x + \delta_{rob}, y)$ and $||\delta_{rob}|| < \epsilon_{rob}$, where ϵ_{rob} is a perturbation budget.

Poison robustness (PoisnRob) protects against *poisoning* which involves training f on *poisons* which are obtained by either tampering existing data records in $\mathcal{D}_{tr}$ or adding manipulated data records to $\mathcal{D}_{tr}$ to

[1]Code will be publicly available upon publication.

degrade ϕ_u (Tian et al., 2022). Alternatively, poisoning for backdoors forces f to incorrectly learn a mapping of some pattern in the poisons, to a target class chosen by $\mathcal{A}dv$. During inference, any data record with that pattern is then misclassified to the target class (Li et al., 2022).

Model Watermarking (MdlWM) checks for *unauthorized model ownership*, including *model extraction attacks* where $\mathcal{A}dv$ trains a local *surrogate model* to mimic the functionality of f (Orekondy et al., 2019). MdlWM embeds watermarks in f that transfer to the surrogate model during extraction. If a suspect model's watermark accuracy is above some pre-defined threshold, it is identified as a surrogate.

Fingerprinting (Fngrprnt) also checks for *unauthorized model ownership* by generating unique identifiers or *fingerprints* (e.g., adversarial examples, embeddings), for f. These fingerprints transfer from f to any surrogate model that are derived from it but are distinct from the fingerprints of independently trained models (Cao et al., 2021; Peng et al., 2022; Lukas et al., 2021; Zheng et al., 2022c; Maini et al., 2021).

Data watermarking (DtWM) checks for *unauthorized data use* where f is trained on datasets collected without consent (e.g., face images for facial recognition) (Sablayrolles et al., 2020; Huang et al., 2021; Wenger et al., 2023). DtWM either augments $\mathcal{D}_{tr}$ with watermarks (e.g., backdoors) (Tekgul & Asokan, 2022; Sablayrolles et al., 2020), or selects high-influence samples from $\mathcal{D}_{tr}$ as watermarks (Liu et al., 2022a). For verification, we check whether watermarks were in $\mathcal{D}_{tr}$ using statistical tests (Sablayrolles et al., 2020) or membership inference (Liu et al., 2022a). The difference between MdlWM and DtWM is how a model trained from scratch on $\mathcal{D}_{tr}$ is classified: DtWM flags it for unauthorized data use while MdlWM classifies it as independently trained.

Differential privacy (DiffPriv) protects against membership inference (whether a data record was in $\mathcal{D}_{tr}$) (Hu et al., 2022) and data reconstruction (reconstructing data records in $\mathcal{D}_{tr}$) (Fredrikson et al., 2015) by hiding whether an individual's data record was used to train f (Abadi et al., 2016). Given two models trained on neighboring datasets differing by one record, DiffPriv bounds the privacy loss (distinguishability in predictions between the two models) by $e^{\epsilon}_{dp} + \delta$. Here, e^{ϵ}_{dp} is the privacy budget and δ_{dp} is probability where the privacy loss is $> e^{\epsilon}_{dp}$.

Group fairness (GpFair) minimizes *discriminatory behavior* to ensure equitable behavior across demographic groups identified by a sensitive attribute in x (e.g., race or sex) (Mehrabi et al., 2021; Pessach & Shmueli, 2022). GpFair is measured using various metrics like accuracy parity, demographic parity (Zafar et al., 2019), equalized odds and equality of opportunity (Hardt et al., 2016).

Explanations (Expl) give insights into f's *incomprehensible behavior* (Guidotti et al., 2018) which can be used to detect *discriminatory behavior* (Selvaraju et al., 2017; Kim et al., 2018). Explanations $\gamma(x)$ indicate the influence of different input attributes in x on $f(x)$. There are three main categories: *Attribution-based* (Ismail et al., 2021; Smilkov et al., 2017; Sundararajan et al., 2017); *influence-based* (Koh & Liang, 2017); and *recourse-based* (Wachter et al., 2017). We focus on attribution-based explanations which are popular in prior work on combining defenses, and applicable to various domains (e.g., tabular, image). These explanations require training a linear model in a region around a point of interest x (Ismail et al., 2021; Smilkov et al., 2017; Sundararajan et al., 2017). The coefficients of the linear model for an input $x = (x_1, \cdots x_n)$ with n attributes, constitutes $\gamma(x)$.

We summarize the defenses and their impact on ϕ_u in Table 8.

3.2 Combination Techniques

Based on our survey described later in §4, we identify two combination techniques which either *modify existing defenses*, or identify whether *existing defenses can be combined without modification*. We mark them as **T1** and **T2** respectively, and describe them as follows:

T1 (Optimization) includes game-theoretic formalization, regularization, or constrained equation solving. **T1** incorporates defenses into the objective function (e.g., regularization terms) so that the corresponding defense constraints can be satisfied during training for an effective combination (Xin et al., 2023; Hu et al., 2023a; Bu et al., 2022; Wu et al., 2023; Zhang & Bu, 2022; He et al., 2020; Tran et al., 2022; Ali Mousavi et al., 2023; Benz et al., 2021; Ma et al., 2022; Nanda et al., 2021; Xu et al., 2021b; Sun et al., 2022; Li

& Liu, 2023; Lee et al., 2024; Wei et al., 2023; P & Abraham, 2021; Liu et al., 2021; Shekhar et al., 2021; Zhang & Davidson, 2021; Tran et al., 2021b; Liu et al., 2022b; Lowy et al., 2023; Jagielski et al., 2019; Tran et al., 2021a; Yaghini et al., 2024; Ding et al., 2020; Xu et al., 2019a; Zhang et al., 2021; Esipova et al., 2023; Xu et al., 2020; Tran et al., 2023; Lakkaraju et al., 2020; Chen et al., 2019; Li et al., 2023b). This also includes using variants of standard model architectures and algorithms, specifically catered for a particular combination to give better trade-offs among the defenses (Ding et al., 2020; Xu et al., 2019a; Yang et al., 2022; Phan et al., 2019; 2020).

T2 (Mutually Exclusive Placement) consists of applying defenses at *different* stages of the ML pipeline— i) pre-training (modifies $\mathcal{D}_{tr}$), ii) in-training (modifies training configuration such as objective function), iii) post-training (modifies inputs or outputs of trained f during inference)—to avoid conflicts (Yaghini et al., 2023; Patel et al., 2022). We later refer to **T2** as the *naïve technique* and use it as a baseline to compare with our proposed technique DEF\CON (§5.3 and §7).

3.3 Type of Interactions

Consider two defenses $\texttt{D}_1$ and $\texttt{D}_2$ which protect against risks $\texttt{Rk}_1$ and $\texttt{Rk}_2$ respectively, with $\texttt{D}_2$ is applied after $\texttt{D}_1$. The can interact in one of two ways:

- **Alignment.** $\texttt{D}_1$ and $\texttt{D}_2$ are *aligned* if any of these hold: (i) $\texttt{D}_1$ and $\texttt{D}_2$ do not impact $\texttt{Rk}_2$ and $\texttt{Rk}_1$, respectively; (ii) $\texttt{D}_1$ reduces $\texttt{Rk}_2$, increasing $\texttt{D}_2$'s effectiveness; (iii) $\texttt{D}_1$ generalizes $\texttt{D}_2$, so its effectiveness implies that of $\texttt{D}_2$. Alignment leads to an effective defense combination. When one defense implies the other (case (iii)), applying the first may be sufficient since we get the second for free (e.g., attribute privacy and group fairness (Aalmoes et al., 2022)).
- **Conflict.** $\texttt{D}_1$ and $\texttt{D}_2$ *conflict* if any of the following hold: (i) $\texttt{D}_1$ uses risk $\texttt{Rk}$ (protected by $\texttt{D}_2$), making $\texttt{D}_1$ ineffective; (ii) $\texttt{D}_2$ overrides $\texttt{D}_1$'s changes, making $\texttt{D}_1$ ineffective. Conflict leads to an ineffective combination of the defenses. To avoid conflicts, we need accurate combination techniques.

3.4 Completeness of Framework

In §3.1, we identify some ML defenses for analysis. We do not claim that this list is complete. For instance, there are other defenses (e.g., *individual fairness* (Dwork et al., 2012), and *interpretability* (Kleinberg & Mullainathan, 2019)), or defenses specific to models other than classifiers (e.g., language and diffusion models), and settings (e.g., federated learning). We can update the framework (§3.1) to add new defenses and enumerate all its combinations with other defenses (as shown later in Table 1). Similarly, in §3.2, we do not claim that the list of combination techniques is complete, but it covers all the techniques seen in our systematization (§4). New combination techniques can be easily added into our framework, and later used for systematization as shown in §4.1.

4 Systematizing Interactions among Defenses

We now use our framework to categorize existing literature. We present our methodology (§4.1), and show the systematization of prior work (§4.2).

4.1 Methodology

We enumerate all defense combinations from §3.1, in Table 1. Each combination is represented as a cell in Table 1, for which we indicate related work, combination technique used, and the type of resulting interaction.

For each prior work, we identify the following:
Combination Technique: We mark the technique to combine defenses as **T1** or **T2**.
Type of Interaction: We mark the interactions as a conflict (Ξ), alignment (Ξ), or unexplored (Ξ).

Justification. Prior surveys are limited to specific defenses (e.g., Chen et al. (2023); Fioretto et al. (2022); Noppel & Wressnegger (2024)), or do not cover sufficient details to about combination techniques (e.g., Gittens et al. (2022); Ferry et al. (2023)). This makes it challenging to design better combination

techniques, and systematically compare with prior works. Our systematization addresses these limitations by (a) covering multiple defenses and their combinations, and (b) explicitly mapping them to the combination techniques and the type of resulting interactions. As shown later in §5, our systematization helps to identify gaps in existing literature (e.g., unexplored combinations, and limitations of prior techniques). Using the insights from our systematization, we can design and evaluate a new combination technique (§5.3 and §7).

Selecting Papers for Analysis. We started surveying papers in Google Scholar using keywords (e.g, "combining <defense 1> and <defense 2>"). We selected all papers including those published in top-tier ML and security/privacy venues (e.g., NeurIPS, ICML, ICLR, AAAI, CCS, S&P), related workshop papers, and unpublished papers on ArXiv. We examined their citations and related work to find other papers. Finally, we used papers from related surveys (e.g., Gittens et al. (2022); Chen et al. (2023); Fioretto et al. (2022); Noppel & Wressnegger (2024); Ferry et al. (2023)) to ensure a comprehensive coverage.

Table 1: **Overview of Pairwise Combinations among Defenses**: For each combination cell, we citep related work and indicate the "interaction type" (Ξ → alignment, Ξ → conflict, Ξ → unexplored), and "combination technique" used (**T1**-**T2**).

	EvsnRob	**PoisnRob**	**MdlWM**	**Fngrprnt**	**DtWM**	**DiffPriv**	**GpFair**
PoisnRob	Ξ→ **T1**: (Xin et al., 2023; Hu et al., 2023a)						
MdlWM	Ξ→ **T2**: (Szyller & Asokan, 2023) Ξ→**T2**: (Thakkar et al., 2023)	Ξ					
Fngrprnt	Ξ→ **T2**: (Szyller & Asokan, 2023) Ξ→ **T2**: (Lukas et al., 2021)	Ξ	Ξ				
DtWM	Ξ→ **T2**: (Szyller & Asokan, 2023)	Ξ	Ξ	Ξ			
DiffPriv	Ξ→ **T1**: (Bu et al., 2022; Wu et al., 2023; Phan et al., 2019; 2020; Zhang & Bu, 2022; He et al., 2020)	Ξ→ **T2**: (Xu et al., 2021a; Vos et al., 2023; Ma et al., 2019)	Ξ→ **T2**: (Szyller & Asokan, 2023)	Ξ→ **T2**: (Szyller & Asokan, 2023)	Ξ→ **T2**: (Szyller & Asokan, 2023)		
GpFair	Ξ: **T1**: (Tran et al., 2022; Ali Mousavi et al., 2023; Benz et al., 2021; Ma et al., 2022; Nanda et al., 2021; Xu et al., 2021b; Sun et al., 2022; Li & Liu, 2023; Lee et al., 2024; Wei et al., 2023) Ξ→ **T2**: (Sun et al., 2022)	Ξ→**T1**: (P & Abraham, 2021; Liu et al., 2021; Shekhar et al., 2021; Zhang & Davidson, 2021)	Ξ	Ξ	Ξ	Ξ→ **T1**: (Tran et al., 2021b; Liu et al., 2022b; Lowy et al., 2023; Jagielski et al., 2019; Tran et al., 2021a; Yaghini et al., 2024; Ding et al., 2020; Xu et al., 2019a; Zhang et al., 2021; Esipova et al., 2023; Xu et al., 2020; Tran et al., 2023) Ξ→ **T2**: (Yaghini et al., 2023)	
Expl	Ξ→ **T1**: (Lakkaraju et al., 2020; Chen et al., 2019; Li et al., 2023b)	Ξ	Ξ	Ξ	Ξ	Ξ→ **T1**: (Yang et al., 2022); **T2**: (Patel et al., 2022)	Ξ

4.2 Survey of Prior Work

We describe the defense combinations in the order of appearance along the columns in Table 1.

EvsnRob + PoisnRob. EvsnRob suppresses adversarial examples (as outliers) while PoisnRob suppresses poisons (as outliers) in $\mathcal{D}_{tr}$. Hence, their objectives are aligned. Viewing PoisnRob as out-of-distribution (OOD) generalization, we can modify adversarial training by incorporating noise from the new domain to improve domain generalization (Xin et al., 2023). This allows the model to learning robust features for OOD generalization, thereby aligning with PoisnRob (Ξ: **T1**). Hu et al. (2023a) defend against both poisons and evasion using a bi-level optimization (Ξ: **T1**).

EvsnRob + MdlWM. Adversarial training, as EvsnRob.In, suppresses the influence of backdoors which are used for MdlWM.Pre (Ξ: **T2**) (Szyller & Asokan, 2023). However, generating adversarial-example based watermarks with a higher ϵ_{rob} than EvsnRob.In, can result in an effective combination (Ξ: **T2**) (Thakkar et al., 2023).

EvsnRob + DtWM. Radioactive data (Sablayrolles et al., 2020), as DtWM.Pre, adds backdoors as watermarks to $\mathcal{D}_{tr}$ by perturbing the inputs (similar to adversarial examples). Hence, adversarial training will suppress the influence of watermarks used for DtWM (Ξ: **T2**) (Szyller & Asokan, 2023).

EvsnRob + Fngrprnt. Dataset inference (Maini et al., 2021) (as Fngrprnt) is effective with EvsnRob and incurs an acceptable performance drop (Ξ: **T2**) (Szyller & Asokan, 2023). We attribute this to the defenses being applied at different stages (in-training vs. post-training), which reduces conflict between them. On the other hand, a variant of Fngrprnt based on adversarial examples (i.e., "conferrable examples"), are ineffective when EvsnRob is applied for the target or the surrogate model (Ξ: **T2**) (Lukas et al., 2021). We mark them both separately in Table 1.

EvsnRob + DiffPriv. Hayes et al. (2022) show that the generalization is worse on combining the objectives of adversarial training (as EvsnRob.In) and DPSGD (as DiffPriv.In), suggesting a conflict (Ξ: **T1**).

Bu et al. (2022) modify the minimax objective function of adversarial training to include DPSGD without violating its guarantees (Ξ: **T1**). Wu et al. (2023) combine randomized smoothing with DPSGD by averaging the gradients of multiple training sample augmentations before clipping to account for the privacy budget of adversarial examples. Both techniques modify the objective function (Ξ: **T1**).

Training f on some public data along with the choice of DiffPriv hyperparameters followed by task specific fine-tuning can result in better trade-off (Ξ: **T1**) (Zhang & Bu, 2022; He et al., 2020). Other works have considered different optimizations: add DiffPriv noise to both input and hidden layers, ensemble adversarial training to add adversarial examples to private $\mathcal{D}_{tr}$, or modify objective function for DiffPriv guarantees on adversarial examples (Ξ: **T1**) (Phan et al., 2020; 2019).

EvsnRob + GpFair. Adversarial training (as EvsnRob.In) and group fairness (as GpFair.In) have conflicting objectives: EvsnRob.In pushes the decision boundary away from $\mathcal{D}_{tr}$ while GpFair.In brings it closer (Tran et al., 2022). Also, EvsnRob.In increases the disparity among demographic subgroups due to class imbalance in $\mathcal{D}_{tr}$ (Hu et al., 2023c) and long-tailed distribution (Lee et al., 2024; Benz et al., 2021; Nanda et al., 2021; Hu et al., 2023c). Several works modify EvsnRob.In's objective function to ensure equitable model behavior across demographic subgroups by assigning higher weight to minority subgroup (Ξ: **T1**) (Ali Mousavi et al., 2023; Benz et al., 2021; Ma et al., 2022; Nanda et al., 2021; Xu et al., 2021b; Sun et al., 2022; Lee et al., 2024; Li & Liu, 2023). Wei et al. (2023) use different training configurations and assigning different weights to different classes to improve class-wise robustness (Ξ: **T1**).

EvsnRob + Expl. Adversarial training (as EvsnRob.In) improves the interpretability of the gradients (Tsipras et al., 2019). This suggesting an alignment with explanations (Chalasani et al., 2020). Also, both defenses can be combined using a minimax objective to construct high fidelity explanations while resisting adversarial examples (Ξ: **T1**) (Lakkaraju et al., 2020; Chen et al., 2019; Li et al., 2023b).

PoisnRob + DiffPriv. DiffPriv reduces the influence of outliers thereby improving robustness against poisons as shown in several works (Xu et al., 2021a; Vos et al., 2023; Ma et al., 2019; Jagielski & Oprea,

2021). Hence, DPSGD mitigates poisons and the defenses have aligned objective, resulting in an effective combination (Ξ: **T2**).

PoisnRob + GpFair. PoisnRob may overly flag data records from the minority groups as outliers for removal, which increases the bias (Shekhar et al., 2021). This can be corrected by reweighing the scores assigned to outliers to account for sensitive attributes (Ξ: **T1**) (P & Abraham, 2021; Liu et al., 2021). Also, an outlier detector can be trained to minimize the correlation between outlier scores and sensitive attributes (Ξ: **T1**) (Shekhar et al., 2021; Zhang & Davidson, 2021).

MdlWM + DiffPriv. DPSGD (as DiffPriv.In) reduces memorization of data records in $\mathcal{D}_{tr}$ and reduces the impact of backdoors for watermarking (MdlWM). Hence, MdlWM conflicts with DiffPriv (Ξ: **T2**) (Szyller & Asokan, 2023).

DtWM + DiffPriv. Ideally, DPSGD (as DiffPriv.In) suppresses watermarks (in DtWM), suggesting a conflict. However, empirically, DtWM was effective when combined with DPSGD (Szyller & Asokan, 2023). The adversarial example-based watermarks radioactive watermarking (Sablayrolles et al., 2020), were relatively inliers and not suppressed by DPSGD (Ξ: **T2**) (Szyller & Asokan, 2023).

Fngrprnt + DiffPriv. Szyller and Asokan (Szyller & Asokan, 2023) found that DPSGD (as DiffPriv.In) and dataset inference (Fngrprnt) do not conflict, though no reason was provided. We attribute this to defenses being applied in different stages, reducing conflict (Ξ: **T2**).

DiffPriv + GpFair. DPSGD (as DiffPriv.In) shows disparate behavior over demographic subgroups (Bagdasaryan et al., 2019). Theoretically, it is impossible to design a high utility binary classifier that is private and fair (Cummings et al., 2019; Agarwal, 2021). Several works modify the objective function by using fairness constraints, regularization, and game theoretic optimization (Ξ: **T1**) (Tran et al., 2021b; Liu et al., 2022b; Lowy et al., 2023; Tran et al., 2021a; Jagielski et al., 2019; Yaghini et al., 2024; Mozannar et al., 2020). Yaghini et al. (2023) combine demographic parity regularization with DPSGD, and estimate fairness on a public dataset to avoid consuming extra privacy budget (Ξ: **T1**). Also, functional mechanism adds Laplace noise to the objective function, along with varied noise levels for different subgroups, which reduces discrimination (Ξ: **T1**) (Ding et al., 2020; Xu et al., 2019a). However, this is limited to the convex objective functions (e.g., logistic regression). Esipova et al. (2023) attribute unfairness in DPSGD to the differences in unclipped and clipped gradient directions. Subsequently, several works have used proposed variable gradient clipping to minimize discriminatory behavior while maintaining utility (Ξ: **T1**) (Xu et al., 2020; Tran et al., 2023; Zhang et al., 2021). Yaghini et al. (2023) use PATE framework to apply fairness constraints and DiffPriv noise in the aggregated votes from the teacher's ensemble. Both fairness and privacy are applied in pre-training (Ξ: **T2**).

DiffPriv + Expl. The objectives of these defenses are inherently conflicting: DiffPriv hides information to minimize leakage while Expl releases additional information to improve comprehensibility (Banisar, 2011). Yang et al. (2022) train an autoencoder with DiffPriv (functional mechanism). This autoencoder is used to generate data records and compute counterfactuals that satisfy DiffPriv via the post-processing property (Ξ: **T1**). Patel et al. (2022) propose an adaptive DPSGD algorithm to generate high-quality explanations without consuming ϵ_{dp}, by reusing past explanations for similar data records (Ξ: **T2**).

5 Insights from Systematization

We identify unexplored combinations (§5.1), requirements for an ideal technique, limitations of prior techniques (§5.2), and design a new technique (§5.3).

5.1 Unexplored Combinations

We identify 14 unexplored combinations (Ξ in Table 1): (i) PoisnRob with {MdlWM, DtWM, Fngrprnt, Expl}; (ii) MdlWM with {DtWM, Fngrprnt, GpFair, Expl}; (iii) DtWM with {Fngrprnt,GpFair,Expl}; (iv) Fngrprnt with {GpFair, Expl}; (v) GpFair with Expl.

Takeaway: Unexplored combinations reveal research gaps and opportunities for effective technique design.

We revisit these combinations in our evaluation (§6 and §7).

5.2 Evaluating Combination Techniques

Our systematization reveals that some techniques may lead to ineffective combinations. We outline requirements for an ideal technique, and identify limitations in prior work.

Requirements. A combination technique should alow practitioner to quickly determine whether a combination can be effective combined. Empirical evaluation to determine the effectiveness of a combination, while definitive, can be expensive, especially when multiple defenses are involved. An ideal combination technique should be: **R1 (Accurate)** correctly identifies whether a combination is effective or not; **R2 (Scalable)** allows two or more to be combined simultaneously; **R3 (Non-invasive)** does not require modifying defenses, easing adoption and removing the need for expert knowledge; **R4 (General)** applicable to various defenses.

Takeaway: Combination techniques, including newly proposed ones, should be evaluated for **R1**-**R4**.

Limitations of Prior Techniques. We summarize the limitations of existing techniques (**T1**-**T2**) as per the requirements (**R1**-**R4**) in Table 2. We use ○ for requirement not satisfied, ◐ for partially satisfied, and ● for fully satisfied.

Table 2: Requirements satisfied by various techniques: ○ → Not satisfied; ◐ → Partially satisfied; ● → fully satisfied.

Technique	R1 (Accurate)	R2 (Scalable)	R3 (Non-Invasive)	R4 (General)
T1	◐	○	○	○
T2	◐	●	●	●

T1 (Optimization) where modifying the objective function for training, followed by hyperparameter tuning, can result in an effective combination. However, this often results in a trade-off between effectiveness of constituent defenses and model utility. Hence, we mark **T1** as partially accurate (**R1** → ◐). This trade-off also explains why prior works have struggled to scale beyond two defenses (**R2** → ○). Furthermore, some defenses are not applicable for **T1** (e.g., Expl and Fngrprnt), and require modifications or non-standard variants (**R3** → ○). Finally, these optimizations are tailored to specific defenses being combined, and do not apply to other defenses. They are also specific to some models (e.g., logistic regression with DiffPriv), and do not translate to other models (e.g., neural networks). Hence, **T1** has limited applicability (**R4** → ○).

T2 (Mutually Exclusive Placement) can apply up to one defense in each of the three stages, thus, making it scalable (**R2** → ●). Defenses do not need any modification (**R3** → ●), and the combination technique is applicable to all types of defenses (**R4** → ●). However, the combinations may not be effective: (i) a defense in a later stage of the pipeline can conflict with earlier ones (false negatives) (Szyller & Asokan, 2023), and (ii) it rules out defenses in the same stage that do not conflict (false positives see §7). Hence, this may incorrectly identify effective combinations (partially accurate **R1** → ◐).

Takeaway. *Neither technique satisfies all the requirements.* **T2** is promising as it satisfies **R2**, **R3**, and **R4**, but not **R1** (incurs false positives and false negatives).

5.3 Def\Con: Design

From our systematization, we identify that **T2** overlooks underlying causes for conflicts, leading to false positives/negatives. *Can we address the limitations of **T2** and improve **R1**?* Recall from §3 that conflicts arise when (i) an early-stage defense uses a risk which is mitigated by a later-stage defense, or (ii) changes by an early-stage defense is overridden by a later-stage defense. We conjecture that by accounting for these underlying causes, we can meet **R1**. We present Def\Con, a principled technique to identify effective defense combinations, by accounting for the reasons underlying conflicts

Methodology to Derive Def\Con. We start with the naïve technique and modify it to include the underlying causes for conflicts among defenses. We iterate over the design of Def\Con using prior work from §4, and evaluate the final design on unexplored combinations (see §7).

Def\Con Description. We describe Def\Con using the example of combining two defenses, $\texttt{D}_1$ and $\texttt{D}_2$ which protect against $\texttt{Rk}_1$ and $\texttt{Rk}_2$ respectively, and later discuss how to extend to more than two defenses. Following prior work (Duddu et al., 2024a), we refer to unintended interactions between a defense and a risk if the defense either increases or decreases the susceptibility to an unrelated risk (e.g., $\texttt{D}_1$ and $\texttt{Rk}_2$).

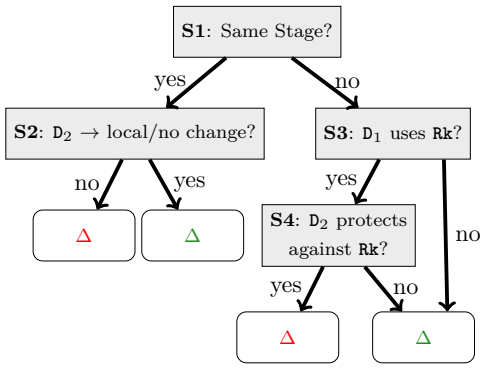

Figure 1: Flowchart depicting various steps in Def\Con to identify conflict between $\texttt{D}_1$ and $\texttt{D}_2$ ($\texttt{D}_2$ is applied after $\texttt{D}_1$).

We start by identifying variants of each of the defenses across pre-, in-, and post-training stages (see Table 8). We compare each variant of $\texttt{D}_1$ with that of $\texttt{D}_2$, and use Δ for alignment, and Δ for conflict. Assuming $\texttt{D}_1$ is applied first and then $\texttt{D}_2$, we follow the steps below *in sequence*:

S1 Are $\texttt{D}_1$ and $\texttt{D}_2$ applied in the same stage?
• If yes, go to **S2** • Else, go to **S3**

S2 The type of changes made by the defenses determines whether there is a conflict. We classify the changes as *global*, *local*, and *none*. *Global changes* modify f (e.g., training with a regularization term, pruning) or transform all records in $\mathcal{D}_{tr}$ (e.g., synthetic data generation for DP or fairness during pre-training). *Local changes* affect specific data records (e.g., adding watermarks in pre-training or modifying certain predictions in post-training). Fngrprnt.Post and Expl.Post make *no changes* to f and $\mathcal{D}_{tr}$.

- If $\texttt{D}_1$ makes global/local/no changes while $\texttt{D}_2$ makes local/no changes, we mark this as Δ.
  Rationale: *Changes by $\texttt{D}_1$ will not interfere with local/no changes by $\texttt{D}_2$, as $\texttt{D}_1$ is applied first. Hence, no conflict.*
- If $\texttt{D}_1$ makes global/local/no changes while $\texttt{D}_2$ makes global changes, mark as Δ.
  Rationale: *Global changes by $\texttt{D}_2$ will override changes by $\texttt{D}_1$, thereby reducing its effectiveness. This is called catastrophic forgetting when the defenses are applied sequentially during training (Kemker et al., 2018; Szyller & Asokan, 2023). This is a conflict.*

S3 $\texttt{D}_1$ and $\texttt{D}_2$ are in different stages. Does $\texttt{D}_1$ use a risk $\texttt{Rk}$ as part of the defense (e.g., watermarking uses backdoors)?

- If yes, go to **S4**.
- If no, mark as Δ.
  Rationale: *If $\texttt{D}_1$ does not use $\texttt{Rk}$, the susceptibility to $\texttt{Rk}$ will not be impacted after applying $\texttt{D}_2$. Hence, $\texttt{D}_1$ and $\texttt{D}_2$ are unlikely to interfere with each other.*

S4 Does $\texttt{D}_2$ protect against $\texttt{Rk}$ either explicitly or via unintended interaction?

- If yes, mark as Δ.
  Rationale: *Since $\texttt{D}_1$ uses $\texttt{Rk}$ (either explicitly or via unintended interaction), $\texttt{D}_2$ will reduce susceptibility to $\texttt{Rk}$ making $\texttt{D}_1$ less effective. Hence, there is a conflict.*
- If no, mark as Δ.
  Rationale: *$\texttt{D}_1$ and $\texttt{D}_2$ are unlikely to interfere with each other. Hence, there is no conflict.*

We summarize the steps in Def\Con in Figure 1. Def\Con evaluates combination effectiveness based solely on the effectiveness of the constituent defenses, without considering the model utility. We revisit model utility in §8. We also present a formal analysis of Def\Con covering consistency, soundness, and completeness in Appendix B.

Note on DiffPriv. Combining DiffPriv with other defenses does not consume additional privacy budget. Any modification to $\mathcal{D}_{tr}$ (e.g., adding watermarks) is done within the privacy boundary, and does not consume privacy budget. Any defense applied after DiffPriv is "free" (DiffPriv's post-processing property).

Extending Beyond Two Defenses (Multi-way Combinations). To extend Def\Con beyond two defenses, our algorithm decomposes the multi-way combination into pairwise comparisons, invoking the original Def\Con flowchart (Figure 1) to check for conflicts. Determining conflict for multi-way combinations from pairwise combinations is principled and not limiting. We have to consider two possible cases when decomposing multi-way combinations into pairwise combinations:

- **Case 1: Defenses in different stages.** When we combine three defenses in different stages, checking conflicts among all pairwise combinations is fine since the order in which we apply defenses is fixed. Any conflict detected among any pair, will be marked as an overall conflict (marked as Δ).
- **Case 2: More than one defense in one stage.** We have to check all possible permutations of the defenses in a given stage, and determine whether there is a conflict. If there is no conflict, we combine with other stage defenses, and check for the conflicts.

We now present our algorithm to evaluate conflict for multi-way combinations, and assume that we have a set of defenses—partitioned into three ordered stages. We iterate through each stage in order, and skip stages with no defenses:

M1 If stage has single defense: For the stage s, we consider the defense as $\texttt{D}_s$ and check its compatibility with defenses in other stages. $\texttt{D}_s$ is $\texttt{D}_1$, $\texttt{D}_2$ or $\texttt{D}_3$ depending on s.

M2 If stage has multiple defenses: For each stage with multiple defenses:

M2.1 Permutation Checking: Consider all permutations of the defenses. For each permutation, check every consecutive pair of defenses using Def\Con flowchart to detect conflicts for two defenses.

M2.2 Pruning and Selection: If a pair in a permutation causes a conflict, discard that permutation and prune others containing that conflicting pair (e.g., for defenses: A, B, and C, if "AB" conflict, then no need to check for "ABC" or "CAB"). If a conflict-free permutation is found, treat the entire sequence as a single composite defense, $\texttt{D}_s$ (e.g., let us say "CAB" does not conflict).

M3 Sequential Composition: Treat the resulting composite defense as an atomic unit (e.g., $\texttt{D}_1$ = "CAB") and check for conflicts with the next stage's resolved defenses (e.g., $\texttt{D}_2$ = "DE"), using Def\Con flowchart between $\texttt{D}_1$ and $\texttt{D}_2$. In other words, we check whether any of the constituent defenses in $\texttt{D}_1$ use a risk which is mitigated by constituent defenses in $\texttt{D}_2$ (**S3** in Figure 1).

M4 Termination: If any invocation of Def\Con for pairwise check indicates a conflict, terminate and report as a conflict (marked as Δ). Else, indicate as alignment (marked as Δ).

We present a formal analysis of the algorithm to extend Def\Con for more than two defenses in Appendix C. Having discussed the Def\Con's design, we comprehensively evaluate Def\Con across **R1**-**R4**.

6 Experimental Setup

To evaluate Def\Con across different requirements (**R1**-**R4**), we present the dataset and metrics used (§6.1), identify defenses in different stages of ML pipeline (§6.2), select defenses for evaluation (§6.3), and present evaluation metrics and our implementation (§6.4).

6.1 Datasets and Models

We use two image datasets: `FMNIST` and `UTKFACE`. `FMNIST` consists of 28x28 grayscale images of ten clothing types, with 60,000 training and 10,000 testing images. We classify these using a two layer CNN with 16 and 32 filters, ReLU activation, and a fully connected layer for ten-class classification. `UTKFACE` includes 48x48 RGB images, classifying individuals as young (under 30), with 11,852 training and 10,667 testing images. It also includes the sex of the individuals as a sensitive attribute. We use a VGG16 model with a fully connected layer for binary classification.

We choose `FMNIST` since all of the defenses we consider for evaluation (§6.3) have used it for evaluation. We selected `UTKFACE` because many defenses effective for `FMNIST` are likely to be applicable to it as well, given that both are image datasets. Also, `UTKFACE` includes sensitive attributes, making it suitable for GpFair.

6.2 Revisiting Defenses

Since the naïve technique and DEF\CON apply defenses at different ML pipeline stages, we revisit and categorize defenses in §3.1 by the stage they are applied in. For each defense from §3.1, we specify the variants in pre-training ("<defense>.**Pre**"), in-training ("<defense>.**In**"), and post-training ("<defense>.**Post**"). For additional context, we indicate the impact of applying a defense on ϕ_{u} compared to a "no defense" baseline, where "$\vee$" is a decrease, "$\sim$" is no effect, and "$\wedge$" is an increase.

Evasion robustness (EvsnRob)

- **EvsnRob.Pre (Data Augmentation)** where adding transformations of training data records improves robustness (Yun et al., 2019; Zhang et al., 2018b; DeVries & Taylor, 2017; Rebuffi et al., 2021) (but see §6.3). This improves ϕ_{u} ($\wedge$) by acting as regularization (Yun et al., 2019; Zhang et al., 2018b; DeVries & Taylor, 2017; Rebuffi et al., 2021).
- **EvsnRob.In (Adversarial Training)** modifies the objective function to minimize the maximum loss from adversarial examples (Madry et al., 2018; Zhang et al., 2019): $L_{advtr} = \min_\theta \frac{1}{|\mathcal{D}_{tr}|} \sum_{x,y \in \mathcal{D}_{tr}} \max_{\|\delta\| \leq \epsilon_{rob}} \ell(f(x+\delta), y)$. Alternatively, randomized smoothing modifies the training and inference to obtain certified robustness of f (Cohen et al., 2019; Lecuyer et al., 2019). These defenses decrease ϕ_{u} ($\vee$) (Zhang et al., 2019; Tsipras et al., 2019).
- **EvsnRob.Post (Input Processing)** removes adversarial perturbations before passing them to f (e.g., using generative models (Nie et al., 2022; Song et al., 2018) or input encoding (Buckman et al., 2018; Guo et al., 2018; Das et al., 2017)) or checks for adversarial examples using statistical tests (Grosse et al., 2017). Defenses which modify input images using generative models decrease ϕ_{u} ($\vee$) (Nie et al., 2022; Song et al., 2018; Guo et al., 2018; Das et al., 2017). If the input transformation is small, the decrease in ϕ_{u} is negligible ($\sim$) (Buckman et al., 2018; Grosse et al., 2017).

Poison robustness (PoisnRob)

- **PoisnRob.Pre (Data Sanitization)** includes detecting and removing outliers in $\mathcal{D}_{tr}$ (e.g., using Shapley values (Jia et al., 2021b; 2019; Doan et al., 2020) or anomaly detection (Cretu et al., 2008; Paudice et al., 2018; Tran et al., 2018; Barreno et al., 2010; Chen et al., 2018)), followed by retraining. As the outliers are memorized and contribute to ϕ_{u}, their removal degrades ϕ_{u} ($\vee$) (Jia et al., 2021b; 2019).
- **PoisnRob.In (Fine-tuning)** updates f to minimize outlier influence. This includes distillation to reduce the influence of poisons (Li et al., 2017) or fine-tuning on a poison-free dataset (Diakonikolas et al., 2019; Zhu et al., 2023; Xu et al., 2019b; Liu & Guo, 2020; Patrini et al., 2017). These do not impact ϕ_{u} ($\sim$).
- **PoisnRob.Post (Pruning)** reduces the effectiveness of backdoors by removing some model parameters based on the observation that poisoned and clean samples have different activations (Liu et al., 2018; Wu & Wang, 2021; Zheng et al., 2022b;a; Li et al., 2023c). This degrades ϕ_{u} ($\vee$).

Model Watermarking (MdlWM)

- **MdlWM.Pre (Backdoors)** uses backdoor watermarks in $\mathcal{D}_{tr}$ (Adi et al., 2018; Zhang et al., 2018c; Jia et al., 2021a; Uchida et al., 2017). These are designed to maintain ϕ_{u} ($\sim$).
- **MdlWM.In (Optimization)** updates the original objective function to include watermark behavior (Bansal et al., 2022; Bagdasaryan & Shmatikov, 2021). For instance, certified watermarking adds Gaussian noise to watermarks (added to $\mathcal{D}_{tr}$) to get certification on watermark accuracy (Bansal et al., 2022). Also, backdoors can be introduced through regularization, which can be repurposed for watermarking (Bagdasaryan & Shmatikov, 2021). This degrades ϕ_{u} ($\vee$).
- **MdlWM.Post (API)** modifies predictions to embed watermarks (Szyller et al., 2021) which are used by $\mathcal{A}dv$ to train the surrogate model. These are designed to maintain ϕ_{u} ($\sim$).

Fingerprinting (Fngrprnt). All fingerprints are post-training schemes (denoted as **Fngrprnt.Post**). No retraining or modification of f is required and hence, FNGRPRNT has no effect on ϕ_{u} ($\sim$).

Data watermarking (DtWM). All the current schemes are during pre-training (**DtWM.Pre**), and are designed to maintain ϕ_{u} ($\sim$). The difference between MDLWM and DTWM is how a model trained from scratch on $\mathcal{D}_{tr}$ is classified: DTWM flags it for unauthorized data use while MDLWM classifies it as independently trained.

Differential privacy (DiffPriv)

- **DiffPriv.Pre (Private Data)** from generative models with DiffPriv constraints, that can be used for downstream tasks instead of $\mathcal{D}_{tr}$ (Hu et al., 2023b; Xie et al., 2018; Torkzadehmahani et al., 2019; Zheng & Li, 2023). This decreases ϕ_{u} ($\vee$).
- **DiffPriv.In (DPSGD)** trains f by adding carefully computed noise to the gradients to minimize the influence of individual data records on f (Abadi et al., 2016). Private aggregation of teacher's ensembles (PATE) (Papernot et al., 2017) is another framework for DP where multiple teacher models are trained on disjoint private datasets, while a student model is trained on a public dataset with labels annotated via noisy voting from the teacher models. These defenses decrease ϕ_{u} ($\vee$) (Jayaraman & Evans, 2019).
- **DiffPriv.Post (Output Perturbation)** includes adding calibrated noise to the output of empirical risk minimization objective (Chaudhuri et al., 2011). This decreases ϕ_{u} ($\vee$). The theoretical guarantees are poorer than other DP defenses and DiffPriv.Post requires the objective function to be convex. We omit this since it does not cover neural networks.

Group fairness (GpFair)

- **GpFair.Pre (Fair Data)** modifies $\mathcal{D}_{tr}$ to reduce bias in the downstream model (Kamiran & Calders, 2011; Calmon et al., 2017; Zemel et al., 2013; Feldman et al., 2015). This degrades ϕ_{u} ($\vee$).
- **GpFair.In (Regularization)** penalizes violating fairness constraints (Agarwal et al., 2018; 2019; Celis et al., 2019; Kamishima et al., 2012). This degrades ϕ_{u} ($\vee$) (Zhang et al., 2018a; Louppe et al., 2017; Pinzón et al., 2023).
- **GpFair.Post (Calibration)** adjusts the threshold over the predictions to ensure that the prediction probabilities accurately reflect the true likelihood across each demographic group (Pleiss et al., 2017; Hardt et al., 2016; Kamiran et al., 2012; Geyik et al., 2019; Salvador et al., 2022; Kull et al., 2017; Hebert-Johnson et al., 2018) This degrades ϕ_{u} ($\vee$) (Pleiss et al., 2017).

Explanations (Expl). Expl are post-training defenses (**Expl.Post**) which does not require retraining, and hence does not degrade ϕ_{u} ($\sim$).

We summarize the defenses in Appendix A: Table 8.

6.3 Choosing Defenses for Evaluation

To select defenses for our evaluation, we began with those in §6.2 (summarized in Appendix A: Table 8). We remove defenses which are not robust: EvsnRob.Post (Input Processing) and PoisnRob.Pre (Data Sanitization) (Kang et al., 2024; Koh et al., 2022). We then evaluate the remaining defenses and exclude those which were ineffective on our datasets: EvsnRob.Pre (Data Augmentation) (Yun et al., 2019; Zhang et al., 2018b; DeVries & Taylor, 2017), DiffPriv.Pre (Private Data) (Zheng & Li, 2023), GpFair.Pre (Fair Data) (Zemel et al., 2013), and GpFair.Post (Calibration) (Pleiss et al., 2017). DiffPriv.Pre, GpFair.Pre, and GpFair.Post, were designed for tabular datasets but ineffective on our image datasets. We speculate about these defenses in §8.

We are left with eleven defenses: (i) EvsnRob.In (adversarial training), (ii) PoisnRob.In (fine-tuning), (iii) PoisnRob.Post (model pruning), (iv) MdlWM.Pre (backdoor watermarks), (v) MdlWM.In (watermarks via objective function), (vi) MdlWM.Post (API-based watermarks), (vii) DtWM.Pre (backdoor watermarks), (viii) Fngrprnt.Post (dataset inference), (ix) DiffPriv.In (DPSGD), (x) GpFair.In (regularization), (xi) Expl.Post (attribution). We get 55 pairwise combinations from them but we remove combinations among defenses with the same objective: three combinations among watermarking (MdlWM.Pre, MdlWM.In, MdlWM.Post), three for Fngrprnt.Post with MdlWM.Pre, MdlWM.In, MdlWM.Post, and one for PoisnRob.In and PoisnRob.Post. This leaves us with 48 combinations.

6.4 Metrics and Implementations

We describe the metrics for evaluating the effectiveness of each defense, and the implementations taken from publicly available code from prior work. We measure ϕ_{u} on $\mathcal{D}_{te}$ for all defenses. Our implementations for

defenses are based on the state-of-the-art (FNGRPRNT, POISNROB), standard libraries (DIFFPRIV, GPFAIR, EXPL), or seminal work (EVSNROB, DTWM, MDLWM). We use the standard hyperparameters which are either from the literature or the library documentation, such that the resulting individual defenses are effective (Table 3). When combining defenses, we use the same hyperparameters, but revisit hyperparameter tuning for defenses in combination (see §7.4). We report the mean and standard deviation across five runs.

Evasion Robustness (EvsnRob.In). We use the accuracy on $\mathcal{D}_{rob}$ which is obtained by replacing data records in $\mathcal{D}_{te}$ with the adversarial variants:

$$\phi_{\text{robacc}}(f_{\text{EVSNROB}}, \mathcal{D}_{\text{rob}}) = \frac{1}{|\mathcal{D}_{\text{rob}}|} \sum_{(x,y)\in\mathcal{D}_{\text{rob}}} \mathbb{I}\left\{\hat{f}_{\text{EVSNROB}}(x) = y\right\}$$

Ideally, $\phi_{\text{robacc}}^{\text{EVSNROB}}$ should be close to ϕ_{u}. We use the original implementation of TRADES (Zhang et al., 2019) and an implementation of AutoAttack (Croce & Hein, 2020) in SecML library (Pintor et al., 2022b) from Pintor et al. (2022a). As we evaluate effectiveness of EVSNROB.IN using attacks, poorly optimized attacks can falsely suggest defense effectiveness (Carlini et al., 2019; Carlini & Wagner, 2017; Tramer et al., 2020). We individually optimize these attacks for evaluation with defenses and their combinations, following Pintor et al. (2022a) to address failures identified by various indicators (e.g., poor optimization). We evaluate across various AutoAttack variants by (a) modifying the loss function: cross entropy (CE) and difference of logits ratio (DLR), (b) applying expectation over transformations (EoT), and (c) using random starts. For `FMNIST`, we use DLR, CE, DLR+EoT, and CE+EoT. For `UTKFACE`, we use CE and CE+EoT, as DLR is not applicable for binary classification. We report the best attack (least ϕ_{robacc}).

Outlier Removal (PoisnRob). We compute accuracy on $\mathcal{D}_{bd}$ (from adding backdoors to records in $\mathcal{D}_{te}$):

$$\phi_{\text{ASR}}(f_{\text{POISNROB}}, \mathcal{D}_{\text{bd}}) = \frac{1}{|\mathcal{D}_{\text{bd}}|} \sum_{(x,y)\in\mathcal{D}_{\text{bd}}} \mathbb{I}\left\{\hat{f}_{\text{POISNROB}}(x) = y\right\}$$

where y_t is the target label chosen by $\mathcal{A}dv$. Ideally, ϕ_{ASR} should be zero. We use BadNets (Gu et al., 2017) to generate poisons by adding a white patch of size 5x5 to the images, applied to 10% of $\mathcal{D}_{tr}$. For POISNROB.IN (Fine-tuning), we fine-tune the last layers of f using random sample of 10% of $\mathcal{D}_{tr}$ without poisons (Sha et al., 2022). For POISNROB.POST (Pruning), we use the implementation from Zheng et al. (2022b). We sweep pruning thresholds from 0.6 to 1.5, in increments of 0.05, to get a model with highest ϕ_{u} and lowest ϕ_{ASR}.

Model-Watermarking (MdlWM). We use the accuracy on $\mathcal{D}_{\text{wmM}}$ which is obtained by adding watermarks to data records in $\mathcal{D}_{te}$. We compute this *watermark accuracy* as:

$$\phi_{\text{wmacc}}(f_{\text{MDLWM}}, \mathcal{D}_{\text{wmM}}) = \frac{1}{|\mathcal{D}_{\text{wmM}}|} \sum_{(x,y)\in\mathcal{D}_{\text{wmM}}} \mathbb{I}\left\{\hat{f}_{\text{MDLWM}}(x) = y\right\}$$

where y_m represents the target labels for watermarked records. Ideally, ϕ_{wmacc} should be 100% if the model is successfully watermarked. For MDLWM.PRE (Backdoor), we use BadNets (Gu et al., 2017), similar to Szyller and Asokan (Szyller & Asokan, 2023), by adding a white patch of size 5x5 to 10% of the images in $\mathcal{D}_{tr}$. For MDLWM.IN (Modifying Loss), we use the certified neural network watermarking implementation by Bansal et al. (2022). For MDLWM.POST (API), we use DAWN (Szyller et al., 2021), which flips a fraction of the predictions from target model as watermarks, which is later used to train the surrogate model. Following the original work (Szyller et al., 2021), we apply the watermark to 0.2% of the predictions. Unlike other watermarking schemes, we compute ϕ_{wmacc} on the surrogate model and not the target model.

Fingerprinting (Fngrprnt.Post). We use dataset inference (Maini et al., 2021) as our fingerprinting scheme which extracts feature embeddings from f, and trains a classifier to distinguish between $\mathcal{D}_{tr}$ and $\mathcal{D}_{te}$. A model is considered stolen if the distance of its embeddings is similar to f with high confidence, and verification is successful if the p-value < 0.05. We use ϕ_{pval} as the metric following Szyller and Asokan (Szyller & Asokan, 2023). We use the step size of 1.0 for L_1 attack, 0.01 for L_2 attack, and 0.001 for L_{inf}, and 50 samples for computing p-value from the confidence regressor model.

Data-Watermarking (DtWM.Pre). To determine if a dataset was used to train a model, we compare the posterior probability of 100 watermarked testing samples against 100 benign ones using a pairwise t-test (Li et al., 2020). We then calculate the rate of successful detection (ϕ_{rsd}), which reflects the percentage of correctly identified watermarked samples from $\mathcal{D}_{\text{wmD}}$ ($\mathcal{D}_{te}$ with watermarks). Watermarks are generated using BadNets (Gu et al., 2017) where 10% of $\mathcal{D}_{tr}$ is watermarked, and we use verification code from Li et al. (2020) to compute ϕ_{rsd}. Ideally, ϕ_{rsd} should be 100% for watermarked models.

Differential Privacy (DiffPriv.In). We use $\phi_{\text{dp}} = \epsilon_{dp}$, following Szyller and Asokan (Szyller & Asokan, 2023), where ideally, we want a low ϵ_{dp}. We use the implementation from Opacus library (Yousefpour et al., 2021) with a noise multiplier of 1.0 and gradient norm clipping of 1.0 as used in their tutorial for `MNIST`.

Group Fairness (GpFair.In). We measure fairness using the *equalized odds gap* on $\mathcal{D}_{te}$ for sensitive attributes S and model predictions $\hat{Y}$, given by:

$$\phi_{\text{eqodd}} = P(\hat{Y} = \hat{y} \mid S = 0, Y = y) \quad - P(\hat{Y} = \hat{y} \mid S = 1, Y = y)$$

$\forall (\hat{y}, y) \in \{0, 1\}^2$ where an ideal value of zero indicates perfect fairness. For GpFair.In (Regularization), we use code from the fair fairness benchmark that adds a regularization term to penalize equalized odds violations (Han et al., 2023). We set the regularization hyperparameter $\lambda = 1$ which was sufficient to reduce ϕ_{eqodds} with $\sim 2\%$ drop in ϕ_{u}.

Explanations (Expl.Post). We assess the quality fo explanations using *convergence delta*, measures the error between the explanation for a data records and a baseline (Kokhlikyan et al., 2020). We report the average convergence delta across all $\mathcal{D}_{te}$ records as ϕ_{err}. We use DeepLift (Shrikumar et al., 2017) from Captum library which recommends using a zero vector as a baseline.

7 Evaluation

We evaluate individual defenses (§7.1), compare Def\Con to naïve technique (§7.2 and §7.3), study the impact of hyperparameter tuning (§7.4), and confirm that Def\Con meets all requirements (§7.5).

7.1 Evaluating Individual Defenses

We evaluate the effectiveness of each defense by comparing the metrics $\phi^{\mathtt{D}}_{(.)}$ to a "no defense" baseline. We report the results in Table 3. We also report ϕ_{u} to provide context but do not use it to evaluate accuracy of the technique.

We find that all the defense effectiveness metrics are better than the "no defense" baseline. Once the defenses are applied, we use their respective $\phi^{\mathtt{D}}_{(.)}$ as the "single defense" baseline to compare the effectiveness of the defense combinations later in §7.3. For $\phi^{\text{Expl.Post}}_{\text{err}}$, we do not have a "no defense" baseline to compare with. Assuming $\phi^{\text{Expl.Post}}_{\text{err}}$ is effective, we use it as the "single defense" baseline.

Table 3: **Effectiveness of defenses.** For metrics, we use ↑ (resp. ↓) where higher (lower) value is better, and "`x`" is shorthand for $\phi^{(.)}_{(x)}$. (ϕ_{u} is for context).

Defense	Metric	FMNIST	UTKFACE
No Defense	**u** (↑)	90.97 ± 0.18	80.28 ± 1.26
	robacc (↑)	7.96 ± 1.24	0.00 ± 0.00
	ASR (↓)	99.95 ± 0.04	99.98 ± 0.05
	wmacc.Pre (↑)	9.98 ± 0.28	0.00 ± 0.00
	wmacc.In (↑)	6.28 ± 1.20	62.21 ± 6.03
	wmacc.Post (↑)	0.00 ± 0.00	13.33 ± 6.32
	RSD (↑)	0.00 ± 0.00	0.00 ± 0.00
	eqodds (↓)		28.10 ± 6.34
	dp (↓)	∞	∞
EvsnRob.In	**u** (↑)	86.93 ± 0.23	73.38 ± 1.15
	robacc (↑)	76.59 ± 0.28	37.38 ± 1.30
PoisnRob.In	**u** (↑)	89.38 ± 0.28	79.02 ± 0.30
	ASR (↓)	9.94 ± 0.24	56.62 ± 37.83
PoisnRob.Post	**u** (↑)	86.48 ± 2.35	65.42 ± 3.27
	ASR (↓)	66.44 ± 21.30	8.59 ± 16.41
MdlWM.Pre	**u** (↑)	90.15 ± 0.27	79.79 ± 0.39
	wmacc (↑)	99.91 ± 0.05	100.00 ± 0.00
MdlWM.In	**u** (↑)	80.87 ± 0.88	66.71 ± 10.19
	wmacc (↑)	85.61 ± 2.50	93.74 ± 11.00
MdlWM.Post	**u** (↑)	90.56 ± 0.34	80.82 ± 0.45
	wmacc (↑)	100.00 ± 0.00	78.10 ± 9.33
DtWM.Pre	**u** (↑)	90.31 ± 0.27	79.93 ± 0.37
	RSD (↑)	100.00 ± 0.00	100.00 ± 0.00
Fngrprnt.Post	**u** (↑)	No change	No change
	pval (↓)	< 0.05	< 0.05
DiffPriv.In	**u** (↑)	86.82 ± 0.11	74.07 ± 0.28
	dp (↓)	$\epsilon_{dp} = 1.36$	$\epsilon_{dp} = 2.89$
GpFair.In	**u** (↑)		76.85 ± 1.99
	eqodds (↓)		10.89 ± 2.84
Expl.Post	**u** (↑)	No change	No change
	err (↓)	0.12 ± 0.03	0.59 ± 0.05

7.2 Accuracy: using Prior Work

Before empirically evaluating 48 defense combinations, we first identify the combinations which have been empirically evaluated in prior work (§4 and Table 5). We identify eight combinations (**C1**-**C8**) whose

results can be used as ground truth to compare the predictions of Def\Con and the naïve technique (marked as Ξ or Ξ in Table 5). We use green and red to indicate alignment and conflict among defenses, respectively. The prediction from a technique is accurate when Δ (or Δ) for Def\Con, or Ψ (or Ψ) for naïve technique, match Ξ (or Ξ) taken from prior work (§4) as ground truth. We present additional details to make predictions in **S2**, **S3**, and **S4** using Def\Con in Table 4.

- **C1 (GpFair.Pre + DiffPriv.Pre)** can be combined effectively in the pre-training stage (Ξ) (Yaghini et al., 2023). Naïve technique predicts Ψ (same stage) while Def\Con predicts Δ (defenses make local changes in **S2**).
- **C2 (EvsnRob.In +Fngrprnt.Post)** can be effectively combined (Ξ) (Szyller & Asokan, 2023). Naïve technique predicts Ψ (different stages) while Def\Con predicts Δ (**S3**=no).
- **C3 (DiffPriv.In + Fngrprnt.Post)** can be effectively combined (Ξ) (Szyller & Asokan, 2023). Naïve technique predicts Ψ (different stages) while Def\Con predicts Δ (**S3**=no).
- **C4 (MdlWM.Pre +EvsnRob.In)** are not effectively combined (Ξ) (Szyller & Asokan, 2023). Naïve technique predicts Ψ (different stages) while Def\Con predicts Δ (EvsnRob.In makes poisons ineffective via unintended interaction in **S4**).
- **C5 (DtWM.Pre +EvsnRob.In)** cannot be effectively combined (Ξ) (Szyller & Asokan, 2023). Naïve technique predicts Ψ (different stages) while Def\Con predicts Δ (EvsnRob.In makes poisons ineffective via unintended interaction in **S4**).
- **C6 (MdlWM.Pre + DiffPriv.In)** cannot be effectively combined (Ξ) (Szyller & Asokan, 2023). Similar to **C5**, Def\Con predictions this as Δ while the naïve technique predicts Ψ.
- **C7 (DtWM.Pre + DiffPriv.In)** can be effectively combined (Ξ) (Szyller & Asokan, 2023). Naïve technique predicts Ψ (different stages) while Def\Con predicts Δ (DiffPriv.In reduces effectiveness of poisons via unintended interaction in **S4**). Unlike backdoor-based watermarks used in our work, adversarial example-based watermarks used by Szyller and Asokan (Szyller & Asokan, 2023), are inliers which are not suppressed by DiffPriv.In. Hence, Def\Con's prediction differs.
- **C8 (DiffPriv.In + Expl.Post)** can be effectively combined (Ξ) (Patel et al., 2022). Naïve technique predicts Ψ (different stages) and Def\Con predicts Δ (**S3**=no).

Of the eight combinations,* Def\Con *predicts seven correctly, while naïve technique predicts four. This gives a balanced accuracy of 90% (TP=4, TN=3, FP=0, FN=1) for* Def\Con*, and 40% (TP=4, TN=0, FP=3, FN=1) for the naïve technique.

Table 4: For each defense, we identify key parameters used in evaluating combination effectiveness (Figure 1): type of change in **S2** (Global, Local, or None); if defense uses a risk in **S3** ("Yes" for backdoors or adversarial examples); and if defense protects against risk in **S4**.

Defense	S2	S3	S4
EvsnRob.In	Global	No	Yes
PoisnRob.In	Global	No	Yes
PoisnRob.Post	Global	No	Yes
MdlWM.Pre	Local	Yes	No
MdlWM.In	Global	Yes	No
MdlWM.Post	Local	No	No
DtWM.Pre	Local	Yes	No
Fngrprnt.Post	None	No	No
DiffPriv.In	Global	No	Yes
GpFair.In	Global	No	No
Expl.Post	None	No	No

7.3 Accuracy: via Empirical Evaluation

We now empirically evaluate the remaining, previously unexplored, combinations to obtain the ground truth and then compute the accuracy of the predictions from both techniques. After removing the eight combinations from prior work, we are left with 40 combinations. We also remove ten combinations where both defenses are applied during in-training. Here, both Def\Con and the naïve technique predict Δ and Ψ respectively. To apply both defenses in the in-training phase, they can be modified for an effective combination (marked as **T1** in §3). However, this makes it invasive (violates **R3**). Alternatively, defenses can be combined sequentially (e.g., pre-training on the first defense, then fine-tuning on the second), or by alternating the training of both defenses every few epochs. Since, these are non-standard approaches to apply existing defenses, we leave a comprehensive evaluation of ten combinations as future work. Hence, ***we are left with 30 combinations (C9-C38) for empirical evaluation.***

Predictions from Techniques. Before evaluating 30 combinations, we denote the defenses as $\mathtt{D}_1$ and $\mathtt{D}_2$ based on the order in which they are applied. We obtain predictions from Def\Con and the naïve techniques,

Table 5: **Evaluating combinations ($\hat{\mathtt{D}}$):** For brevity, "`x`" in **Metric** column is shorthand for $\phi^{\hat{\mathtt{D}}}_{(x)}$. We use ↑ (resp. ↓) to indicate if a higher (lower) value is better. For defense effectiveness, we use green when $\phi^{\hat{\mathtt{D}}}_{(.)}$ is better or equal to "single defense" baseline; orange for better or equal to "single defense" but worse than "no defense"; red for worse than "no defense". For technique predictions, we use symbol Δ (resp. Ψ) to refer to Def\Con (naïve technique) and a color code green (resp. red) to indicate alignment (conflict) among defenses. $\mathtt{D}_1$ and $\mathtt{D}_2$ indicate order of applying defenses. (ϕ_{u} is for context).

	Combinations	Metric	FMNIST	UTKFACE		Combinations	Metric	FMNIST	UTKFACE
C9	$\mathtt{D}_1$: **EvsnRob.In** $\mathtt{D}_2$: **MdlWM.Post** (Ψ, Δ)	**u** (↑) **wmacc** (↑) **robacc** (↓)	90.38 ± 0.22 100.00 ± 0.00 80.43±0.85	72.79 ± 0.53 80.95 ± 7.13 42.00 ± 0.49	**C24**	$\mathtt{D}_1$: **MdlWM.Pre** $\mathtt{D}_2$: **Expl.Post** (Ψ, Δ)	**u** (↑) **err** (↓) **wmacc** (↑)	90.18 ± 0.21 0.14 ± 0.04 99.93 ± 0.06	79.76 ± 0.63 0.02 ± 0.03 99.96 ± 0.08
C10	$\mathtt{D}_1$: **PoisnRob.In** $\mathtt{D}_2$: **Fngrprnt.Post** (Ψ, Δ)	**u** (↑) **ASR** (↓) **pval** (↓)	89.50 ± 0.21 9.94 ± 0.22 <0.05	79.25 ± 1.06 56.09 ± 12.98 <0.05	**C25**	$\mathtt{D}_1$: **MdlWM.In** $\mathtt{D}_2$: **Expl.Post** (Ψ, Δ)	**u** (↑) **err** (↓) **wmacc** (↑)	86.94 ± 0.50 0.19 ± 0.07 98.24 ± 0.66	72.16 ± 5.13 0.37 ± 0.18 97.60 ± 3.54
C11	$\mathtt{D}_1$: **PoisnRob.Post** $\mathtt{D}_2$: **Fngrprnt.Post** (Ψ, Δ)	**u** (↑) **ASR** (↓) **pval** (↓)	84.73 ± 1.72 61.36 ± 23.96 <0.05	63.70 ± 3.87 0.02 ± 0.03 <0.05	**C26**	$\mathtt{D}_1$: **DtWM.Pre** $\mathtt{D}_2$: **Expl.Post** (Ψ, Δ)	**u** (↑) **err** (↓) **RSD** (↑)	90.04 ± 0.60 0.10 ± 0.04 100.00 ± 0.00	79.03 ± 1.10 0.54 ± 0.01 100.00 ± 0.00
C12	$\mathtt{D}_1$: **EvsnRob.In** $\mathtt{D}_2$: **Expl.Post** (Ψ, Δ)	**u** (↑) **err** (↓) **robacc** (↑)	87.10 ± 0.21 0.22 ± 0.01 79.00 ± 0.21	73.65 ± 1.21 0.15 ± 0.04 39.27 ± 0.68	**C27**	$\mathtt{D}_1$: **PoisnRob.In** $\mathtt{D}_2$: **Expl.Post** (Ψ, Δ)	**u** (↑) **ASR** (↓) **err** (↓)	89.39 ± 0.24 9.79 ± 0.15 0.06 ± 0.02	78.71 ± 0.20 44.35 ± 30.07 0.47 ± 0.02
C13	$\mathtt{D}_1$: **GpFair.In** $\mathtt{D}_2$: **PoisnRob.Post** (Ψ, Δ)	**u** (↑) **ASR** (↓) **eqodds** (↓)		66.73 ± 3.24 20.21 ± 39.90 2.72 ± 3.20	**C28**	$\mathtt{D}_1$: **PoisnRob.Post** $\mathtt{D}_2$: **Expl.Post** (Ψ, Δ)	**u** (↑) **ASR** (↓) **err** (↓)	84.62 ± 3.56 76.11 ± 15.85 0.08 ± 0.01	63.80 ± 3.37 0.00 ± 0.00 0.15 ± 0.06
C14	$\mathtt{D}_1$: **MdlWM.Pre** $\mathtt{D}_2$: **GpFair.In** (Ψ, Δ)	**u** (↑) **wmacc** (↑) **eqodds** (↓)		79.02 ± 0.40 98.88 ± 2.13 0.00 ± 0.00	**C29**	$\mathtt{D}_1$: **Fngrprnt.Post** $\mathtt{D}_2$: **Expl.Post** (Ψ, Δ)	**u** (↑) **pval** (↓) **err** (↓)	90.56 ± 0.16 <0.05 0.11 ± 0.02	80.42 ± 0.59 <0.05 0.50 ± 0.03
C15	$\mathtt{D}_1$: **GpFair.In** $\mathtt{D}_2$: **MdlWM.Post** (Ψ, Δ)	**u** (↑) **wmacc** (↑) **eqodds** (↓)		76.95 ± 1.94 80.95 ± 0.00 7.87 ± 4.72	**C30**	$\mathtt{D}_1$: **DtWM.Pre** $\mathtt{D}_2$: **Fngrprnt.Post** (Ψ, Δ)	**u** (↑) **pval** (↓) **RSD** (↑)	90.19 ± 0.59 <0.05 100.00 ± 0.00	79.80 ± 0.48 <0.05 100.00 ± 0.00
C16	$\mathtt{D}_1$: **DtWM.Pre** $\mathtt{D}_2$: **GpFair.In** (Ψ, Δ)	**u** (↑) **RSD** (↑) **eqodds** (↓)		78.97 ± 1.21 100.00 ± 0.00 0.00 ± 0.00	**C31**	$\mathtt{D}_1$: **DiffPriv.In** $\mathtt{D}_2$: **MdlWM.Post** (Ψ, Δ)	**u** (↑) **wmacc** (↑) dp (↓)	86.83 ± 0.20 100.00 ± 0.00 $\epsilon = 1.36$	74.62 ± 0.49 79.05 ± 3.81 $\epsilon = 2.89$
C17	$\mathtt{D}_1$: **GpFair.In** $\mathtt{D}_2$: **Fngrprnt.Post** (Ψ, Δ)	**u** (↑) **pval** (↓) **eqodds** (↓)		78.67 ± 1.46 0.68 ± 0.21 7.46 ± 5.43	**C32**	$\mathtt{D}_1$: **DtWM.Pre** $\mathtt{D}_2$: **MdlWM.Post** (Ψ, Δ)	**u** (↑) **RSD** (↑) **wmacc** (↑)	90.24 ± 0.29 100.00 ± 0.00 100.00 ± 0.00	78.94 ± 0.95 100.00 ± 0.00 62.26 ± 3.77
C18	$\mathtt{D}_1$: **GpFair.In** $\mathtt{D}_2$: **Expl.Post** (Ψ, Δ)	**u** (↑) **err** (↓) **eqodds** (↓)		80.52 ± 0.44 0.16 ± 0.06 12.62 ± 4.20	**C33**	$\mathtt{D}_1$: **PoisnRob.Post** $\mathtt{D}_2$: **MdlWM.Post** (Ψ, Δ)	**u** (↑) **wmacc** (↑) **ASR** (↓)	85.09 ± 1.94 100.00 ± 0.00 59.48 ± 24.91	67.09 ± 2.81 73.33 ± 8.83 40.20 ± 28.82
C19	$\mathtt{D}_1$: **PoisnRob.In** $\mathtt{D}_2$: **MdlWM.Post** (Ψ, Δ)	**u** (↑) **wmacc** (↑) **ASR** (↓)	89.53 ± 0.36 100.00 ± 0.00 10.48 ± 0.46	79.00 ± 0.56 69.52 ± 6.46 38.90 ± 38.73	**C34**	$\mathtt{D}_1$: **DtWM.Pre** $\mathtt{D}_2$: **MdlWM.Pre** (Ψ, Δ)	**u** (↑) **wmacc** (↑) **RSD** (↑)	90.31 ± 0.27 99.96 ± 0.0 100.00 ± 0.00	78.53 ± 1.75 100.00 ± 0.00 100.00 ± 0.00
C20	$\mathtt{D}_1$: **MdlWM.Post** $\mathtt{D}_2$: **Expl.Post** (Ψ, Δ)	**u** (↑) **wmacc** (↑) **err** (↓)	90.93 ± 0.18 100.00 ± 0.00 0.11 ± 0.02	80.53 ± 0.23 72.38 ± 3.56 0.55 ± 0.02	**C35**	$\mathtt{D}_1$: **EvsnRob.In** $\mathtt{D}_2$: **PoisnRob.Post** (Ψ, Δ)	**u** (↑) **robacc** (↑) **ASR** (↓)	90.39 ± 0.63 46.00 ± 1.02 79.68 ± 10.25	80.28 ± 0.39 0.00 ± 0.00 0.00 ± 0.00
C21	$\mathtt{D}_1$: **DtWM.Pre** $\mathtt{D}_2$: PoisnRob.In (Ψ, Δ)	**u** (↑) **ASR** (↓) **RSD** (↑)	89.46 ± 0.32 10.18 ± 0.40 0.00 ± 0.00	79.00 ± 0.67 77.39 ± 35.23 80.00 ± 40.00	**C36**	$\mathtt{D}_1$: **MdlWM.Pre** $\mathtt{D}_2$: **PoisnRob.In** (Ψ, Δ)	**u** (↑) **ASR** (↓) **wmacc** (↑)	89.48 ± 0.15 10.18 ± 0.46 10.18 ± 0.46	79.20 ± 0.60 46.92 ± 36.92 46.92 ± 36.92
C22	$\mathtt{D}_1$: **DtWM.Pre** $\mathtt{D}_2$: **MdlWM.In** (Ψ, Δ)	**u** (↑) **wmacc** (↑) **RSD** (↑)	84.45 ± 0.56 89.25 ± 3.48 100.00 ± 0.00	79.88 ± 0.27 99.98 ± 0.03 100.00 ± 0.00	**C37**	$\mathtt{D}_1$: **MdlWM.Pre** $\mathtt{D}_2$: **PoisnRob.Post** (Ψ, Δ)	**u** (↑) **ASR** (↓) **wmacc** (↑)	82.86 ± 4.16 71.32 ± 14.11 71.31 ± 14.10	64.09 ± 3.09 0.00 ± 0.00 0.00 ± 0.00
C23	$\mathtt{D}_1$: **DtWM.Pre** $\mathtt{D}_2$: **PoisnRob.Post** (Ψ, Δ)	**u** (↑) **ASR** (↓) **RSD** (↑)	82.90 ± 2.06 64.55 ± 21.23 80.00 ± 40.00	69.02 ± 1.96 0.01 ± 0.01 20.00 ± 40.00	**C38**	$\mathtt{D}_1$: **MdlWM.In** $\mathtt{D}_2$: **PoisnRob.Post** (Ψ, Δ)	**u** (↑) **ASR** (↓) **wmacc** (↑)	66.68 ± 9.80 58.59 ± 19.22 58.65 ± 19.23	73.69 ± 3.01 99.60 ± 0.37 99.73 ± 0.29

and indicate them as a tuple: (Naïve prediction, Def\Con prediction). These are indicated in Table 5. We use the information in Table 4 to make predictions in **S2**-**S4** for Def\Con.

- For defenses applied in the same stage (**S1**=yes), the naïve technique predicts Ψ. We have the following cases to determine the prediction from Def\Con:
  1. $\mathtt{D}_2$ makes local/no changes (**S2**=no), Def\Con predicts this as Ψ. We mark them as (Ψ, Δ) which include **C11**, **C20**, **C28**, **C29**, **C33**, and **C34**.

2. D_2 makes global changes (**S2**=yes), and Def\Con predicts this as Δ. We mark them as (Ψ, Δ) but we did not observe any such combinations.

- For defenses applied in different stages (**S1**=no), the naïve technique predicts Ψ. We have the following cases to determine the prediction from Def\Con:
  1. D_1 does not use a risk (**S3**=no) and hence, D_1 and D_2 do not conflict. We mark them as (Ψ, Δ): **C9**, **C10**, **C12**, **C13**, **C15**, **C17-C19**, **C27**, **C31**, and **C35**.
  2. D_1, such as MdlWM.Pre and DtWM.Pre, uses a risk (**S3**=yes), but D_2 does not protect against this risk (**S4**=no). Hence, there is no conflict and we mark such combinations as (Ψ, Δ) which include **C14**, **C16**, **C22**, **C24-C26**, **C30**, and **C32**.
  3. D_1, such as MdlWM.Pre and DtWM.Pre, uses a risk (**S3**=yes), and D_2 mitigates these risks (e.g., PoisnRob). There is a conflict and we mark them as (Ψ, Δ) which include **C21**, **C23**, **C36-C38**.

We evaluate the 30 combinations on `FMNIST` and `UTKFACE` (Table 5). For each combination, we compare the effectiveness metrics for each defense to "single defense" from Table 3. We use green to indicate that the metrics are better or similar to "single defense"; orange for worse than single defense but better than "no defense"; and red for similar or worse than "no defense". Metrics marked as orange can still be useful since it provides some protection compared to "no defense". We consider the worst case by a treating a combination as a conflict if atleast one dataset has atleast one metric marked as orange or red.

Of the 30 combinations,* Def\Con *predicts 27 correctly, while the naïve method predicts only 18. This gives a balanced accuracy of 81% (TP=22, TN=5, FP=3, and FN=0) for* Def\Con *compared to 36% (TP=16, TN=0, FP=8, and FN=6) for the naïve technique.

Takeaway: By explicitly accounting for reasons underlying conflicts among defenses, Def\Con achieves higher accuracy than the naïve technique (satisfies **R1**).

7.4 Hyperparameter Tuning for Combinations

We check if hyperparameter tuning can resolve conflicts to see if it can turn (i) **false positives** (predicted as aligned, but empirically conflicting) into **true positives**, and (ii) **true negatives** (predicted and confirmed as conflict) into **false negatives**. We exclude **false negatives** (not observed in our evaluation), and **true positives**, correctly predicted and confirmed as aligned (cannot be improved further with hyperparameter tuning). We use grid search and identify various hyperparameter configurations for defenses in conflicting combinations (Table 6).

Table 6: Configurations for hyperparameter tuning of defenses in conflicting combinations.

Defense	Hyperparameter	Values
EvsnRob.In	Regularization	{2, 4, 6 (default), 8}
PoisnRob.Post	Pruning threshold	0.6–1.5 (step 0.05)
MdlWM.Pre	Trigger size Watermark fraction	{3×3, 5×5 (default)} {0.1 (default), 0.2, 0.3}
MdlWM.In	Watermark fraction Training noise Noise step size	{0.1, 0.2, 0.3} {0.5, 0.75, 1.0 (default), 1.25} {0.05 (default), 0.10, 0.15}
MdlWM.Post	Watermark fraction	{0.002 (default), 0.01, 0.02}
DtWM.Pre	Trigger size Watermark fraction	{3×3, 5×5 (default)} {0.1 (default), 0.2, 0.3}
GpFair.In	Regularization	{0.5, 1 (default), 1.5, 2}
Fngrprnt.Post	Iterations # Fingerprints	{25, 50 (default), 75, 100} {100 (default), 150, 200}

Do False Positives turn to True Positives?
This includes three combinations (**C17**, **C32**, **C35**), and helps investigate Def\Con errors.

- **C17 (GpFair.In + Fngrprnt.Post)**. We empirically observe a conflict as Fngrprnt.Post is ineffective ($\phi_{\text{pval}} > 0.05$), and Def\Con incorrectly predicts the combination as Δ in **S3**. We explore the following hyperparameters: regularization for GpFair.In, iterations, and number of fingerprints for Fngrprnt.Post. For each dataset, we have 36 experiments ($= 4 \times 4 \times 3$). None of the experiments alleviated the conflict. Following prior work (Szyller & Asokan, 2023), we speculate that Fngrprnt.Post is ineffective because it relies on the decision boundary, which shifts significantly after applying GpFair.In.
- **C32 (DtWM.Pre + MdlWM.Post)**. The combination is empirically effective for `FMNIST` but not for `UTKFACE` where ϕ_{wmacc} is less than the "single defense" baseline. Def\Con incorrectly predicts this combination as Δ in **S4**. We explore the following hyperparameters: (i) trigger size and watermark fraction for DtWM.Pre; (ii) watermark fraction for MdlWM.Post. For `UTKFACE`, we evaluate 18 experiments ($= 2 \times 3 \times 3$). One experiment with 3×3 trigger size (DtWM.Pre), 30% watermarks

(DtWM.Pre), and 2% watermarks (MdlWM.Post), we get $\phi_u = 75.98 \pm 0.61$, $\phi_{RSD} = 100.00 \pm 0.00$ (green), and $\phi_{wmacc} = 70.19 \pm 4.61$ (green). Hence, we remove the conflict and the false positive.

- **C35 (EvsnRob.In + PoisnRob.Post)**. Empirically, there is a conflict as EvsnRob.In is ineffective (poor ϕ_{robacc}), and Def\Con incorrectly predicts as Δ in **S4**. We vary the regularization hyperparameter (EvsnRob.In), and pruning thresholds (PoisnRob.Post). None of the experiments removed the conflict. We speculate that the model parameters memorizing poisons and adversarial examples overlap. Thus, pruning a model (to reduce ϕ_{ASR}) trained with EvsnRob.In, also reduces ϕ_{robacc}, resulting in a conflict.

Do True Negatives turn to False Negatives? We evaluate hyperparameter tuning for five combinations (**C21**, **C23**, **C36**, **C37**, and **C38**).

- **C21 (DtWM.Pre + PoisnRob.In)**. We consider trigger size, and watermark fraction for DtWM.Pre. For each dataset, we get six experiments (=2×3). None of them removed the conflict since PoisnRob.In mitigates backdoors for DtWM.Pre.
- **C23 (DtWM.Pre + PoisnRob.Post)**. We tune the same hyperparameters for DtWM.Pre as in **C21**. For PoisnRob.Post, we sweep across various pruning thresholds. For each dataset, we get six experiments (= 2 × 3). For `FMNIST`, trigger size of 3×3 and 30% watermarks, gives ϕ_u =82.00 ± 5.50; ϕ_{RSD}= 100.00 ± 0.00; ϕ_{ASR}= 59.90 ± 12.81. This is marked as no conflict (green). However, there is a conflict for `UTKFACE`, making the overall combination a conflict.
- **C36 (MdlWM.Pre + PoisnRob.In)**. We tune the same hyperparameters for MdlWM.Pre, as DtWM.Pre in **C21**. For each dataset, we have six experiments (=2×3). None of them removed the conflict since PoisnRob.In mitigates backdoors for MdlWM.Pre.
- **C37 (MdlWM.Pre + PoisnRob.Post)**. We tune the same hyperparameters for MdlWM.Pre, as DtWM.Pre in **C21**. For PoisnRob.Post, we sweep across various pruning thresholds. For each dataset, we get six experiments (=2×3). None of them removed the conflict since PoisnRob.Post mitigates backdoors for MdlWM.Pre.
- **C38 (MdlWM.In + PoisnRob.Post)**. We tune the fraction of watermarks, training noise, and step size for MdlWM.In. For PoisnRob.Post, we sweep across various pruning thresholds. For each dataset, we have 36 experiments (=3×4×3). None of them removed the conflict since PoisnRob.Post mitigates the backdoors for MdlWM.In.

Summary. We find that hyperparameter tuning is useful in two combinations (**C23** and **C32**). For **C32**, we removed the false positive, thereby increasing Def\Con's balanced accuracy to 86% (from 81%). For **C23**, we could remove conflict for one of the two datasets, but the combination was still marked as a conflict (no additional false negatives).

Takeaway: Hyperparameter tuning for *conflicting combinations* is important to check if it turns (a) false positives to true positives, or (b) true negatives to false negatives.

Factors for Hyperparameter Tuning Effectiveness. We identify factors affecting the effectiveness of hyperparameter tuning and highlight how these factors apply to some conflicting combinations:

- **Expressiveness of hyperparameters:** Tuning is only effective if there are hyperparameters that directly influence the conflicting interaction. If the interaction is insensitive to changes in a hyperparameter, tuning will not resolve the conflict. This could be the reason for hyperparameter tuning being ineffective for some combinations (**C21**, **C36**, **C37**, and **C38**).
- **Search Space:** If the search space is narrow, tuning may never find the optimal configuration to resolve conflicts. Conversely, a sufficiently broad search space increases the chance of finding a configuration that decouples the objectives. There is a possibility that tuning did not work for some combinations as our search space was not broad enough (e.g., **C17**, **C32**, **C35**).
- **Fundamental Incompatibility:** Some defenses are fundamentally incompatible and cannot be resolved by tuning. We discuss this further below.
- **Optimization Landscape:** If the loss landscape contains many local minima, tuning may not find the optimal configuration to resolve conflicts. A practitioner can try more sophisticated tuning instead of grid search (e.g., randomized search or Bayesian optimization), to see if it resolves conflicts.

Limitations. When tuning fails to resolve a conflict, we cannot conclusively mark the combination as a conflict: despite considering a feasible search space in our evaluation, some conflict-resolving hyperparameters may have been missed given the vast number of possibilities. Also, when a combination is marked as conflict (e.g., "**S4**=Yes" for **C21**, **C36**, **C37**, and **C38**), it suggests an *empirical incompatibility* among defenses but does not imply a conclusive proof or *fundamental incompatibility*. Establishing fundamental incompatibility requires a theoretical analysis which is left as future work.

7.5 Other Requirements

Having shown that the naïve technique does not perform as well as Def\Con, we discuss how Def\Con meets the remaining requirements: scalability (**R2**), non-invasive (**R3**), and generality (**R4**).

Scalability (R2). None of the prior works have considered more than two defenses. Since Def\Con allows for applying defenses in three stages of the ML pipeline, it should theoretically support at least three defenses. To illustrate this, we follow the instructions in §5.3 to extend Def\Con beyond two defenses. We begin with pairwise combinations predicted as effective (marked as Δ in Table 5), which align with empirical evaluation, and then include additional defenses. We consider five combinations with three defenses each, which should be effectively combines (marked as Δ). We report the results in Table 7 and find that it is indeed possible to effectively combine three defenses using Def\Con. *Overall, Def\Con scales to more than two defenses (**R2**).* These are illustrative examples to show that Def\Con is scalable to more than two defenses. Our goal was not an exhaustive evaluation of the large number of multi-way combinations, but only to show that effective multi-way combinations–previously unexplored–are possible. A comprehensive evaluation to identify false positives and negatives, is left as future work.

Table 7: **Scalability (R2) of Def\Con to D_1, D_2, and D_3 (in order).** Color coding and notations are same as in Table 5.

	Combinations	Metric	FMNIST	UTKFACE
C39	D_1: EvsnRob.In D_2: Expl.Post D_3: MdlWM.Post	u (↑) robacc (↑) err (↓) wmacc (↑)	87.38 ± 0.15 79.37 ± 0.29 0.96 ± 0.14 100.00 ± 0.00	74.34 ± 0.72 39.21 ± 0.32 0.17 ± 0.05 73.33 ± 8.83
C40	D_1: PoisnRob.In D_2: Expl.Post D_3: MdlWM.Post	u (↑) ASR (↓) err (↓) wmacc (↑)	89.47 ± 0.24 9.81 ± 0.12 0.06 ± 0.02 100.00 ± 0.00	79.42 ± 0.51 66.74 ± 12.11 0.52 ± 0.04 77.14 ± 11.82
C41	D_1: PoisnRob.Post D_2: Expl.Post D_3: MdlWM.Post	u (↑) ASR (↓) err (↓) wmacc (↑)	89.47 ± 0.24 9.81 ± 0.12 0.06 ± 0.02 100.00 ± 0.00	67.04 ± 3.35 1.85 ± 3.39 0.17 ± 0.10 81.90 ± 7.00
C42	D_1: DtWM.Pre D_2: GpFair.In D_3: Expl.Post	u (↑) wmacc (↑) eqodds (↓) err (↓)		77.53 ± 1.75 100.00 ± 0.00 0.00 ± 0.00 0.01 ± 0.00
C43	D_1: DtWM.Pre D_2: GpFair.In D_3: MdlWM.Post	u (↑) RSD (↑) eqodds (↓) wmacc (↑)		79.17 ± 0.93 100.00 ± 0.00 0.00 ± 0.00 73.33 ± 7.12
C44	D_1: GpFair.In D_2: PoisnRob.Post D_3: Expl.Post	u (↑) eqodds (↓) ASR (↓) err (↓)		69.42 ± 2.09 8.12 ± 4.49 0.13 ± 0.25 0.05 ± 0.02

Non-Invasive (R3). Def\Con extends **T2** and hence, inherits the non-invasive requirement. We use existing defenses proposed in the literature without modifying them, and only adapting them to our datasets. *In summary, Def\Con satisfies **R3**.*

General (R4). Def\Con identifies a conflict based on (a) relative position of the defenses in ML pipeline, and (b) the mechanisms that underlie them (whether one defense uses a risk that is being protected by a later defense). Def\Con does not rely on specific defenses. Hence, Def\Con is likely to be applicable beyond the initial set of defenses we used for evaluating our work (commonly available defenses from the literature). For identifying conflicts of a new defense with others, a practitioner can first identify its position in the ML pipeline, and determine whether it uses a risk defended by a later defense. Since these are independent of any specific defense, Def\Con is applicable for any defense that we can map to Def\Con's flowchart (Figure 1). Furthermore, we select specific defense implementations based on their availability (see §6.4). However, other implementations can be used and should not effect our conclusions.

Takeaway: Def\Con scales beyond two defenses (**R2**), is non-invasive (**R3**), and general (**R4**).

8 Discussion, Conclusions, and Future Work

Note on Model Utility. So far, we have focused only on *effectiveness*, examining how combining defenses impacts the effectiveness of each individual defense. An additional pre-requisite for deploying a defense combination is whether it negatively impacts *model utility*. We can define a defense combination to be *viable* if it is (a) effective and (b) incurs only a minimal utility drop compared to lowest of the "single defense" baseline. In Table 5, we observe that all the combinations which DEF\CON predicted as effective are also viable. For **C15**, **C27**, and **C30**, the utility is worse than the "single defense" baseline. These were already flagged as ineffective. We did not observe any combinations which are effective but not viable.

Extending DEF\CON to predict the viability of the combinations is challenging, since it is unclear how a defense impacts model utility. For instance, there can be the following two cases:

- For both defenses, if utility is either better or similar to the "single defense" baseline, it is likely that the combination will have acceptable utility.
- If the utility degrades for both defenses, the combination is likely to have poor utility and hence non-viable. However, it is also possible (as seen in Table 5) that the utility of the combination does not fall below the "minimum utility for single defenses" baseline, even if some, but not all, constituent defenses fall below their respective "no-defense" baseline. It is unclear what mechanisms account for this phenomenon.

Hence, quantifying the impact of individual defenses on utility is an open problem for several defenses (e.g., adversarial training (Zhang et al., 2019; Tsipras et al., 2019; Yang et al., 2020; Pang et al., 2022; Raghunathan et al., 2020) and differential privacy (Jayaraman & Evans, 2019; Ye et al., 2023; Papernot et al., 2021; Tramèr & Boneh, 2020)), and an area of active research. *Viability* can be included as a requirement in §5.2, and extending DEF\CON for viable combinations is left as future work.

Practical Considerations. We discuss the impact on other considerations such as computational cost and latency. The computational cost (for training) and the latency (for inference) are the sum of the costs and latencies, incurred by the constituent defenses when applied individually.

- Pre-training, in-training, and some post-training (e.g., pruning a model) defenses incur a reasonable one-time cost, assuming the practitioner has basic resources to train ML models (e.g., GPUs). These do not have any impact on the inference-time latency.
- For some post-training defenses, there is no training cost but incur a per-inference latency for transforming the inputs or outputs.

Real-world Impact. Following the experimental setup in prior work, we evaluated DEF\CON in a lab setting and not on real-world models, which was not realistic for our experiments. The real-world impact of successful combinations depends on the individual application. Practitioners have to decide which combination should be applied for a given application: for instance, in credit card approval, robustness, fairness, and explainability are important properties, while in medical diagnosis, privacy may also be essential.

Learning-based Component for Def\Con. We can train a model (e.g., decision tree) to predict the type of interaction for a combination. But this requires a lot more data for training (e.g., features covering various defenses and their combinations), than what we can obtain from the eight combinations in prior work. For the current defenses, our heuristic was sufficient to get a reasonable accuracy. As future work, adding a learning-based component is an interesting direction.

Other Causes Underlying Conflicts. While we identify two possible reasons underlying conflicts among defenses (§3.3), we do not claim this to be complete. There could be other underlying reasons which can be included in DEF\CON to make it more accurate. One possible reasons could be the a choice of different l_p-norm distances for some defenses (e.g., EVSNROB.IN, and adversarial example-based MDLWM.PRE and DTWM.PRE). Prior works have shown that the objectives of obtaining robustness to different l_p-norm bounds are conflicting (Tramèr & Boneh, 2019). In case of combinations, Thakkar et al. (2023) show that choosing different amount of noise for watermarking and adversarial training can remove a conflict. We leave the exploration of additional reasons underlying conflicts as future work.

Other Combination Techniques. Duddu et al. (2024a) (Table 3) systematize unintended interactions among defenses and risks, categorizing them as increasing, decreasing, or unexplored. An alternative naïve technique could reject combinations where one defense increases the risks mitigated by another. However, this is restrictive and discards several non-conflicting combinations (e.g., Expl.Post and MdlWM, EvsnRob.In and Fngrprnt.Post). Since there are several unexplored interactions in their systematization, it is challenging to applying this naïve technique in our context. Hence, this technique is limited to some combinations, and not applicable to all combinations in the current state.

Other Dataset Modalities and Models. We rely on two image datasets previously used by prior work to evaluate most of our defenses, making them a natural starting point. Since Def\Con 's steps are modality-independent, we conjecture that it can be applied similarly to other data types. Evaluating Def\Con on new modalities requires implementing corresponding defenses and models (e.g., language models for text). Then, our methodology can be followed: (a) identifying various risks and corresponding defenses in different stages of the ML pipeline, and (b) use the empirical evaluation of combinations as ground truth to verify the accuracy of Def\Con. This is a substantial undertaking, and hence, left as future work since it may bring out new insights. We tried to extend our existing defense implementations to tabular dataset, and indeed, not all defenses transfer to other data modalities. For example, when adapting our defenses to the CENSUS dataset, only 4 of the 11 implemented defenses were applicable—image-specific techniques like poisoning and watermarking do not apply to tabular data. This gave two valid combinations: (i) evasion robustness + explanations and (ii) group fairness + explanations (others were excluded due to incompatibility). Both combinations aligned with Def\Con 's predictions.

Speculating Combinations with Omitted Defenses. We *speculate* on the omitted defense combinations from §6: EvsnRob.Pre, DiffPriv.Pre, GpFair.Pre, and GpFair.Post. Since EvsnRob.Pre targets adversarial examples and makes local changes to $\mathcal{D}_{tr}$, we expect its combination with other defenses to behave similar to MdlWM.Pre and DtWM.Pre. DiffPriv.Pre and GpFair.Pre make global changes by transforming all data records in $\mathcal{D}_{tr}$ and should be applied before other defenses, as we expect them to avoid conflicts. GpFair.Post makes global changes in post-training stage to all predictions, and the behavior is likely to be similar to PoisnRob.Post which also makes global changes to f in post-training stage. Validating these interactions is left for future work. We indicate some steps for future work to validate our speculation. For defenses omitted due to (i) poor effectiveness: further work is required to design more effective defense variants (e.g., pre-training or post-training evasion robustness). (ii) incompatible datasets: set up experiments for the specific dataset where the defenses work well (e.g., tabular datasets instead of image datasets) and evaluate their combinations. We can then combine with other defenses, and evaluate the combinations. We will release our code to combine new defenses with our existing ones.

Trade-offs among Defenses. The current version of Def\Con only outputs alignment and conflict and does not capture trade-offs. It is not clear how to compare the extent of gains/losses for different defenses, and there is no uniform metric that works across all defenses. For instance, 5% loss in one defense may be much worse than 10% loss in another, but there is no clear consensus on which is better. This may be application dependent and has to be determined a practitioner. Hence, we chose to leave this as future work.

Summary. ML models must be protected against multiple risks simultaneously, requiring effective combination of defenses. We systematize prior work, identify unexplored combinations, and evaluate limitations of prior techniques. Using insights from our systematization, we present a technique, Def\Con, which is more accurate than prior work, does not require modifying defenses, scales to more than two defenses, and applies to various defenses.

Ethical Considerations

We use public datasets and implementations and none of our experiments require IRB approval.

Broader Impact

Protecting ML models from various risks simultaneously is an important problem, especially in high-stakes domains where failures can have serious societal consequences. Our work advances the field of designing trustworthy ML systems by moving beyond individual risk mitigation—common in current research—to developing techniques to protect against multiple risks. We do not directly address "accountability" in this work, other than providing a way for model owners to assess effectiveness of defense combinations before deploying them. But it can be combined with additional mechanisms designed to ensure accountability in ML pipelines (such as Duddu et al. (2024b)) for responsible deployment of defenses.

References

Jan Aalmoes, Vasisht Duddu, and Antoine Boutet. On the alignment of group fairness with attribute privacy. *arXiv preprint arXiv:2211.10209*, 2022.

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *USENIX Security Symposium*, pp. 1615–1631, 2018. URL `https://www.usenix.org/conference/usenixsecurity18/presentation/adi`.

Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *International conference on machine learning*, pp. 60–69. PMLR, 2018.

Alekh Agarwal, Miroslav Dudík, and Zhiwei Steven Wu. Fair regression: Quantitative definitions and reduction-based algorithms. In *International Conference on Machine Learning*, pp. 120–129. PMLR, 2019.

Sushant Agarwal. Trade-offs between fairness and privacy in machine learning. In *IJCAI 2021 Workshop on AI for Social Good*, 2021.

Seyed Ali Mousavi, Hamid Mousavi, and Masoud Daneshtalab. Farmur: Fair adversarial retraining to mitigate unfairness in robustness. In *Advances in Databases and Information Systems*, pp. 133–145, 2023.

Eugene Bagdasaryan and Vitaly Shmatikov. Blind backdoors in deep learning models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 1505–1521, 2021.

Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. *Advances in neural information processing systems*, 32, 2019.

David Banisar. The right to information and privacy: Balancing rights and managing conflicts. *SSRN Electronic Journal*, 03 2011.

Arpit Bansal, Ping-yeh Chiang, Michael J Curry, Rajiv Jain, Curtis Wigington, Varun Manjunatha, John P Dickerson, and Tom Goldstein. Certified neural network watermarks with randomized smoothing. In *International Conference on Machine Learning*, pp. 1450–1465, 2022.

Marco Barreno, Blaine Nelson, Anthony D. Joseph, and J. D. Tygar. The security of machine learning. *Mach. Learn.*, 81(2):121–148, 2010. URL `https://doi.org/10.1007/s10994-010-5188-5`.

Philipp Benz, Chaoning Zhang, Adil Karjauv, and In So Kweon. Robustness may be at odds with fairness: An empirical study on class-wise accuracy. In *NeurIPS 2020 Workshop on Pre-registration in Machine Learning*, pp. 325–342. PMLR, 2021.

Eitan Borgnia, Valeriia Cherepanova, Liam Fowl, Amin Ghiasi, Jonas Geiping, Micah Goldblum, Tom Goldstein, and Arjun Gupta. Strong data augmentation sanitizes poisoning and backdoor attacks without an accuracy tradeoff. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3855–3859. IEEE, 2021.

Zhiqi Bu, Ping Li, and Weijie Zhao. Practical adversarial training with differential privacy for deep learning, 2022. URL https://openreview.net/forum?id=1hw-h1C8bch.

Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=S18Su--CW.

Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. Optimized pre-processing for discrimination prevention. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/9a49a25d845a483fae4be7e341368e36-Paper.pdf.

Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Ipguard: Protecting intellectual property of deep neural networks via fingerprinting the classification boundary. In *ACM Asia Conference on Computer and Communications Security*, pp. 14–25, 2021. URL https://doi.org/10.1145/3433210.3437526.

Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *ACM workshop on artificial intelligence and security*, pp. 3–14, 2017.

Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

L Elisa Celis, Lingxiao Huang, Vijay Keswani, and Nisheeth K Vishnoi. Classification with fairness constraints: A meta-algorithm with provable guarantees. In *Conference on fairness, accountability, and transparency*, pp. 319–328, 2019.

Prasad Chalasani, Jiefeng Chen, Amrita Roy Chowdhury, Xi Wu, and Somesh Jha. Concise explanations of neural networks using adversarial training. In *International Conference on Machine Learning*, volume 119, pp. 1383–1391, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/chalasani20a.html.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3), 2011.

Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728*, 2018.

Huiqiang Chen, Tianqing Zhu, Tao Zhang, Wanlei Zhou, and Philip S. Yu. Privacy and fairness in federated learning: On the perspective of tradeoff. *ACM Comput. Surv.*, 56(2), sep 2023. URL https://doi.org/10.1145/3606017.

Jiefeng Chen, Xi Wu, Vaibhav Rastogi, Yingyu Liang, and Somesh Jha. Robust attribution regularization. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/172ef5a94b4dd0aa120c6878fc29f70c-Paper.pdf.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, volume 97, pp. 1310–1320, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/cohen19c.html.

Gabriela F. Cretu, Angelos Stavrou, Michael E. Locasto, Salvatore J. Stolfo, and Angelos D. Keromytis. Casting out demons: Sanitizing training data for anomaly sensors. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pp. 81–95, 2008.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference of Machine Learning*, 2020.

Rachel Cummings, Varun Gupta, Dhamma Kimpara, and Jamie Morgenstern. On the compatibility of privacy and fairness. In *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*, pp. 309–315, 2019. URL https://doi.org/10.1145/3314183.3323847.

Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Li Chen, Michael E. Kounavis, and Duen Horng Chau. Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression. In *ArXiv abs/1705.02900*, 2017.

Emiliano De Cristofaro. An overview of privacy in machine learning. *arXiv preprint arXiv:2005.08679*, 2020.

Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Jacob Steinhardt, and Alistair Stewart. Sever: A robust meta-algorithm for stochastic optimization. In *International Conference on Machine Learning*, volume 97, pp. 1596–1606, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/diakonikolas19a.html.

Jiahao Ding, Xinyue Zhang, Xiaohuan Li, Junyi Wang, Rong Yu, and Miao Pan. Differentially private and fair classification via calibrated functional mechanism. In *AAAI Conference on Artificial Intelligence*, volume 34, pp. 622–629, 2020.

Bao Gia Doan, Ehsan Abbasnejad, and Damith C Ranasinghe. Februus: Input purification defense against trojan attacks on deep neural network systems. In *Annual Computer Security Applications Conference*, pp. 897–912, 2020.

Vasisht Duddu, Sebastian Szyller, and N Asokan. Sok: Unintended interactions among machine learning defenses and risks. *IEEE Symposium on Security and Privacy*, 2024a.

Vasisht Duddu et al. Laminator: Verifiable ml property cards using hardware-assisted attestations. In *CODASPY*, pp. 317–328, 2024b.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Innovations in Theoretical Computer Science Conference*, pp. 214–226, 2012. URL https://doi.org/10.1145/2090236.2090255.

Maria S. Esipova, Atiyeh Ashari Ghomi, Yaqiao Luo, and Jesse C Cresswell. Disparate impact in differential privacy from gradient misalignment. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=qLOaeRvteqbx.

Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 259–268, 2015.

Julien Ferry, Ulrich Aïvodji, Sébastien Gambs, Marie-José Huguet, and Mohamed Siala. SoK: Taming the Triangle - On the Interplays between Fairness, Interpretability and Privacy in Machine Learning. In *https://hal.science/hal-04359832*, 2023.

Ferdinando Fioretto, Cuong Tran, Pascal Van Hentenryck, and Keyu Zhu. Differential privacy and fairness in decisions and learning tasks: A survey. In *International Joint Conference on Artificial Intelligence*, pp. 5470–5477, 7 2022. URL https://doi.org/10.24963/ijcai.2022/766. Survey Track.

Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *ACM SIGSAC Conference on Computer and Communications Security*, pp. 1322–1333, 2015. URL https://doi.org/10.1145/2810103.2813677.

Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. Fairness-aware ranking in search & recommendation systems with application to linkedin talent search. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2221–2231, 2019.

Alex Gittens, Bülent Yener, and Moti Yung. An adversarial perspective on accuracy, robustness, fairness, and privacy: Multilateral-tradeoffs in trustworthy ml. *IEEE Access*, 10:120850–120865, 2022.

Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*, 2017.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM Comput. Surv.*, 51(5), aug 2018. URL https://doi.org/10.1145/3236009.

Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SyJ7ClWCb.

Xiaotian Han, Jianfeng Chi, Yu Chen, Qifan Wang, Han Zhao, Na Zou, and Xia Hu. Ffb: A fair fairness benchmark for in-processing group fairness methods. *arXiv preprint arXiv:2306.09468*, 2023.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

Jamie Hayes, Borja Balle, and M Pawan Kumar. Learning to be adversarially robust and differentially private. *arXiv preprint arXiv:2201.02265*, 2022.

Fengxiang He, Shaopeng Fu, Bohan Wang, and Dacheng Tao. Robustness, privacy, and generalization of adversarial training. *arXiv preprint arXiv:2012.13573*, 2020.

Ursula Hebert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (Computationally-identifiable) masses. In *International Conference on Machine Learning*, volume 80, pp. 1939–1948, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/hebert-johnson18a.html.

Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S. Yu, and Xuyun Zhang. Membership inference attacks on machine learning: A survey. *ACM Comput. Surv.*, jan 2022. URL https://doi.org/10.1145/3523273.

Shu Hu, Zhenhuan Yang, Xin Wang, Yiming Ying, and Siwei Lyu. Outlier robust adversarial training. *arXiv preprint arXiv:2309.05145*, 2023a.

Yuzheng Hu, Fan Wu, Qinbin Li, Yunhui Long, Gonzalo Munilla Garrido, Chang Ge, Bolin Ding, David Forsyth, Bo Li, and Dawn Song. Sok: Privacy-preserving data synthesis. *arXiv preprint arXiv:2307.02106*, 2023b.

Yuzheng Hu, Fan Wu, Hongyang Zhang, and Han Zhao. Understanding the impact of adversarial robustness on accuracy disparity. In *International Conference on Machine Learning*, volume 202, pp. 13679–13709, 23–29 Jul 2023c. URL https://proceedings.mlr.press/v202/hu23j.html.

Hanxun Huang, Xingjun Ma, Sarah Monazam Erfani, James Bailey, and Yisen Wang. Unlearnable examples: Making personal data unexploitable. In *ICLR*, 2021.

Aya Abdelsalam Ismail, Hector Corrada Bravo, and Soheil Feizi. Improving deep learning interpretability by saliency guided training. In *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=x4zs7eC-BsI.

Matthew Jagielski and Alina Oprea. Does differential privacy defeat data poisoning? *International Conference on Learning Representations*, 2021.

Matthew Jagielski, Michael Kearns, Jieming Mao, Alina Oprea, Aaron Roth, Saeed Sharifi Malvajerdi, and Jonathan Ullman. Differentially private fair learning. In *International Conference on Machine Learning*, volume 97, pp. 3000–3008, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/jagielski19a.html.

Bargav Jayaraman and David Evans. Evaluating differentially private machine learning in practice. In *USENIX Security Symposium*, pp. 1895–1912, 2019. URL https://www.usenix.org/conference/usenixsecurity19/presentation/jayaraman.

Hengrui Jia, Christopher A. Choquette-Choo, Varun Chandrasekaran, and Nicolas Papernot. Entangled watermarks as a defense against model extraction. In *USENIX Security Symposium*, pp. 1937–1954, 2021a. URL https://www.usenix.org/conference/usenixsecurity21/presentation/jia.

Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gurel, Bo Li, Ce Zhang, Costas J Spanos, and Dawn Song. Efficient task-specific data valuation for nearest neighbor algorithms. *arXiv preprint arXiv:1908.08619*, 2019.

Ruoxi Jia, Fan Wu, Xuehui Sun, Jiacen Xu, David Dao, Bhavya Kailkhura, Ce Zhang, Bo Li, and Dawn Song. Scalability vs. utility: Do we have to sacrifice one for the other in data importance quantification? In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8239–8247, 2021b.

Faisal Kamiran and Toon Calders. Data pre-processing techniques for classification without discrimination. *Knowledge and Information Systems*, 33, 10 2011.

Faisal Kamiran, Asim Karim, and Xiangliang Zhang. Decision theory for discrimination-aware classification. In *IEEE 12th International Conference on Data Mining*, pp. 924–929, 2012.

Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-aware classifier with prejudice remover regularizer. In *Machine Learning and Knowledge Discovery in Databases*, pp. 35–50, 2012.

Mintong Kang, Dawn Song, and Bo Li. Diffattack: Evasion attacks against diffusion-based adversarial purification. *Advances in Neural Information Processing Systems*, 36, 2024.

Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International Conference on Machine Learning*, pp. 2564–2572, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/kearns18a.html.

Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. An empirical study of rich subgroup fairness for machine learning. In *Conference on fairness, accountability, and transparency*, pp. 100–109, 2019.

Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *AAAI conference on artificial intelligence*, volume 32, 2018.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pp. 2668–2677. PMLR, 2018.

Jon Kleinberg and Sendhil Mullainathan. Simplicity creates inequity: implications for fairness, stereotypes, and interpretability. In *ACM Conference on Economics and Computation*, pp. 807–808, 2019.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, volume 70, pp. 1885–1894, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/koh17a.html.

Pang Wei Koh, Jacob Steinhardt, and Percy Liang. Stronger data poisoning attacks break data sanitization defenses. *Machine Learning*, pp. 1–47, 2022.

Narine Kokhlikyan et al. Captum: A unified and generic model interpretability library for pytorch. *arXiv preprint arXiv:2009.07896*, 2020.

Meelis Kull, Telmo Silva Filho, and Peter Flach. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In *International Conference on Artificial Intelligence and Statistics*, volume 54, pp. 623–631, 20–22 Apr 2017. URL https://proceedings.mlr.press/v54/kull17a.html.

Himabindu Lakkaraju, Nino Arsov, and Osbert Bastani. Robust and stable black box explanations. In *International Conference on Machine Learning*, pp. 5628–5638. PMLR, 2020.

Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE symposium on security and privacy (SP)*, pp. 656–672. IEEE, 2019.

Hyungyu Lee, Saehyung Lee, Hyemi Jang, Junsung Park, Ho Bae, and Sungroh Yoon. DAFA: Distance-aware fair adversarial training. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=BRdEBlwUW6.

Boqi Li and Weiwei Liu. Wat: improve the worst-class robustness in adversarial training. In *AAAI conference on artificial intelligence*, volume 37, pp. 14982–14990, 2023.

L. Li, T. Xie, and B. Li. Sok: Certified robustness for deep neural networks. In *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 1289–1310, 2023a. URL https://doi.ieeecomputersociety.org/10.1109/SP46215.2023.10179303.

Tang Li, Fengchun Qiao, Mengmeng Ma, and Xi Peng. Are data-driven explanations robust against out-of-distribution data? In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3821–3831, June 2023b.

Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and L. Li. Learning from noisy labels with distillation. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 1928–1936, 2017. URL https://doi.ieeecomputersociety.org/10.1109/ICCV.2017.211.

Yige Li, Xixiang Lyu, Xingjun Ma, Nodens Koren, Lingjuan Lyu, Bo Li, and Yu-Gang Jiang. Reconstructive neuron pruning for backdoor defense. In *International Conference on Machine Learning*, pp. 19837–19854. PMLR, 2023c.

Yiming Li, Ziqi Zhang, Jiawang Bai, Baoyuan Wu, Yong Jiang, and Shu-Tao Xia. Open-sourced dataset protection via backdoor watermarking. *arXiv preprint arXiv:2010.05821*, 2020.

Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Gaoyang Liu, Tianlong Xu, Xiaoqiang Ma, and Chen Wang. Your model trains on my data? protecting intellectual property of training data via membership fingerprint authentication. *IEEE Transactions on Information Forensics and Security*, 17:1024–1037, 2022a.

Haoyu Liu, Fenglong Ma, Shibo He, Jiming Chen, and Jing Gao. Fairness-aware outlier ensemble. *arXiv preprint arXiv:2103.09419*, 2021.

Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International symposium on research in attacks, intrusions, and defenses*, pp. 273–294, 2018.

Wenyan Liu, Xiangfeng Wang, Haikun Zheng, Bo Jin, Xiaoling Wang, and Hongyuan Zha. Mitigating disparate impact on model accuracy in differentially private learning. *Information Sciences*, 616:108–126, 2022b. URL https://www.sciencedirect.com/science/article/pii/S0020025522011392.

Yang Liu and Hongyi Guo. Peer loss functions: Learning from noisy labels without knowing noise rates. In *International conference on machine learning*, pp. 6226–6236. PMLR, 2020.

Gilles Louppe, Michael Kagan, and Kyle Cranmer. Learning to pivot with adversarial networks. *Advances in neural information processing systems*, 30, 2017.

Andrew Lowy, Devansh Gupta, and Meisam Razaviyayn. Stochastic differentially private and fair learning. In *International Conference on Learning Representations*, pp. 86–119. PMLR, 2023.

Nils Lukas, Yuxuan Zhang, and Florian Kerschbaum. Deep neural network fingerprinting by conferrable adversarial examples. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=VqzVhqxkjH1.

Xinsong Ma, Zekai Wang, and Weiwei Liu. On the tradeoff between robustness and fairness. In *Advances in Neural Information Processing Systems*, volume 35, pp. 26230–26241, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/a80ebbb4ec9e9b39789318a0a61e2e43-Paper-Conference.pdf.

Yuzhe Ma, Xiaojin Zhu, and Justin Hsu. Data poisoning against differentially-private learners: Attacks and defenses. *International Joint Conferences on Artificial Intelligence*, 2019.

Gabriel Resende Machado, Eugênio Silva, and Ronaldo Ribeiro Goldschmidt. Adversarial machine learning in image classification: A survey toward the defender's perspective. *ACM Comput. Surv.*, 55(1), nov 2021. URL https://doi.org/10.1145/3485133.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Pratyush Maini, Mohammad Yaghini, and Nicolas Papernot. Dataset inference: Ownership resolution in machine learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=hvdKKV2yt7T.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6), jul 2021. URL https://doi.org/10.1145/3457607.

Hussein Mozannar, Mesrob Ohannessian, and Nathan Srebro. Fair learning with private demographic data. In *International Conference on Machine Learning*, pp. 7066–7075. PMLR, 2020.

Vedant Nanda, Samuel Dooley, Sahil Singla, Soheil Feizi, and John P Dickerson. Fairness through robustness: Investigating robustness disparity in deep learning. In *ACM Conference on Fairness, Accountability, and Transparency*, pp. 466–477, 2021.

Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. In *International Conference on Machine Learning (ICML)*, 2022.

M. Noppel and C. Wressnegger. Sok: Explainable machine learning in adversarial environments. In *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 21–21, 2024. URL https://doi.ieeecomputersociety.org/10.1109/SP54263.2024.00021.

Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Knockoff nets: Stealing functionality of black-box models. In *IEEE/CVF conference on computer vision and pattern recognition*, pp. 4954–4963, 2019.

Deepak P and Savitha Abraham. Fairlof: Fairness in outlier detection. *Data Science and Engineering*, 6, 12 2021.

Tianyu Pang, Min Lin, Xiao Yang, Jun Zhu, and Shuicheng Yan. Robustness and accuracy could be reconcilable by (proper) definition. In *International Conference on Machine Learning*, pp. 17258–17277. PMLR, 2022.

Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HkwoSDPgg.

Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael P. Wellman. Sok: Security and privacy in machine learning. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 399–414, 2018.

Nicolas Papernot, Abhradeep Thakurta, Shuang Song, Steve Chien, and Úlfar Erlingsson. Tempered sigmoid activations for deep learning with differential privacy. In *AAAI Conference on Artificial Intelligence*, volume 35, pp. 9312–9321, 2021.

Neel Patel, Reza Shokri, and Yair Zick. Model explanations with differential privacy. In *ACM Conference on Fairness, Accountability, and Transparency*, pp. 1895–1904, 2022. URL https://doi.org/10.1145/3531146.3533235.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *IEEE conference on computer vision and pattern recognition*, pp. 1944–1952, 2017.

Andrea Paudice, Luis Muñoz-González, Andras Gyorgy, and Emil C Lupu. Detection of adversarial training examples in poisoning attacks through anomaly detection. In *arXiv preprint arXiv:1802.03041*, 2018.

Andrea Paudice, Luis Muñoz-González, and Emil C Lupu. Label sanitization against label flipping poisoning attacks. In *ECML PKDD 2018 Workshops: Nemesis 2018, UrbReas 2018, SoGood 2018, IWAISe 2018, and Green Data Mining 2018, Dublin, Ireland, September 10-14, 2018, Proceedings 18*, pp. 5–15. Springer, 2019.

Zirui Peng, Shaofeng Li, Guoxing Chen, Cheng Zhang, Haojin Zhu, and Minhui Xue. Fingerprinting deep neural networks globally via universal adversarial perturbations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13430–13439, 2022.

Dana Pessach and Erez Shmueli. A review on fairness in machine learning. *ACM Comput. Surv.*, 55(3), feb 2022. URL https://doi.org/10.1145/3494672.

Hai Phan, My T Thai, Han Hu, Ruoming Jin, Tong Sun, and Dejing Dou. Scalable differential privacy with certified robustness in adversarial learning. In *International Conference on Machine Learning*, pp. 7683–7694. PMLR, 2020.

NhatHai Phan, Minh Vu, Yang Liu, Ruoming Jin, Dejing Dou, Xintao Wu, and My T Thai. Heterogeneous gaussian mechanism: Preserving differential privacy in deep learning with provable robustness. *arXiv preprint arXiv:1906.01444*, 2019.

Maura Pintor, Luca Demetrio, Angelo Sotgiu, Ambra Demontis, Nicholas Carlini, Battista Biggio, and Fabio Roli. Indicators of attack failure: Debugging and improving optimization of adversarial examples. In *Advances in Neural Information Processing Systems*, volume 35, pp. 23063–23076, 2022a. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/91ffdc5e2f12436d99914418e38d0a09-Paper-Conference.pdf.

Maura Pintor, Luca Demetrio, Angelo Sotgiu, Marco Melis, Ambra Demontis, and Battista Biggio. secml: Secure and explainable machine learning in python. *SoftwareX*, 18:101095, 2022b. URL https://www.sciencedirect.com/science/article/pii/S2352711022000656.

Carlos Pinzón, Catuscia Palamidessi, Pablo Piantanida, and Frank Valencia. On the incompatibility of accuracy and equal opportunity. *Mach. Learn.*, 113(5):2405–2434, may 2023. URL https://doi.org/10.1007/s10994-023-06331-y.

Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. On fairness and calibration. *Advances in neural information processing systems*, 30, 2017.

Han Qiu, Yi Zeng, Shangwei Guo, Tianwei Zhang, Meikang Qiu, and Bhavani Thuraisingham. Deepsweep: An evaluation framework for mitigating dnn backdoor attacks using data augmentation. In *ACM Asia Conference on Computer and Communications Security*, pp. 363–377, 2021. URL https://doi.org/10.1145/3433210.3453108.

Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John Duchi, and Percy Liang. Understanding and mitigating the tradeoff between robustness and accuracy. *arXiv preprint arXiv:2002.10716*, 2020.

Sylvestre-Alvise Rebuffi, Sven Gowal, Dan Andrei Calian, Florian Stimberg, Olivia Wiles, and Timothy A Mann. Data augmentation can improve robustness. *Advances in Neural Information Processing Systems*, 34:29935–29948, 2021.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Radioactive data: tracing through training. In *International Conference on Machine Learning*, pp. 8326–8335. PMLR, 2020.

Tiago Salvador, Stephanie Cairns, Vikram Voleti, Noah Marshall, and Adam M Oberman. Faircal: Fairness calibration for face verification. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=nRj0NcmSuxb`.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017.

Zeyang Sha, Xinlei He, Pascal Berrang, Mathias Humbert, and Yang Zhang. Fine-tuning is all you need to mitigate backdoor attacks. *arXiv preprint arXiv:2212.09067*, 2022.

Shubhranshu Shekhar, Neil Shah, and Leman Akoglu. Fairod: Fairness-aware outlier detection. In *AAAI/ACM Conference on AI, Ethics, and Society*, pp. 210–220, 2021.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pp. 3145–3153. PMlR, 2017.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *ArXiv*, abs/1706.03825, 2017. URL `https://api.semanticscholar.org/CorpusID:11695878`.

Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=rJUYGxbCW`.

Haipei Sun, Kun Wu, Ting Wang, and Wendy Hui Wang. Towards fair and robust classification. In *IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 356–376, 2022.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pp. 3319–3328, 2017.

Sebastian Szyller and N Asokan. Conflicting interactions among protection mechanisms for machine learning models. In *AAAI Conference on Artificial Intelligence*, volume 37, pp. 15179–15187, 2023.

Sebastian Szyller, Buse Gul Atli, Samuel Marchal, and N. Asokan. Dawn: Dynamic adversarial watermarking of neural networks. In *ACM International Conference on Multimedia*, pp. 4417–4425, 2021. URL `https://doi.org/10.1145/3474085.3475591`.

Buse Gul Atli Tekgul and N Asokan. On the effectiveness of dataset watermarking in adversarial settings. *arXiv preprint arXiv:2202.12506*, 2022.

Janvi Thakkar, Giulio Zizzo, and Sergio Maffeis. Elevating defenses: Bridging adversarial training and watermarking for model resilience. *arXiv preprint arXiv:2312.14260*, 2023.

Zhiyi Tian, Lei Cui, Jie Liang, and Shui Yu. A comprehensive survey on poisoning attacks and countermeasures in machine learning. *ACM Comput. Surv.*, 55(8), dec 2022. URL `https://doi.org/10.1145/3551636`.

Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. Dp-cgan: Differentially private synthetic data and label generation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

Florian Tramèr and Dan Boneh. *Adversarial training and robustness for multiple perturbations*. 2019.

Florian Tramèr and Dan Boneh. Differentially private learning needs better features (or much more data). In *International Conference on Learning Representations*, 2020.

Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *Advances in neural information processing systems*, 33:1633–1645, 2020.

Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *International Conference on Neural Information Processing Systems*, pp. 8011–8021, 2018.

Cuong Tran, My Dinh, and Ferdinando Fioretto. Differentially private empirical risk minimization under the fairness lens. *Advances in Neural Information Processing Systems*, 34:27555–27565, 2021a.

Cuong Tran, Ferdinando Fioretto, and Pascal Van Hentenryck. Differentially private and fair deep learning: A lagrangian dual approach. In *AAAI Conference on Artificial Intelligence*, volume 35, pp. 9932–9939, 2021b.

Cuong Tran, Keyu Zhu, Ferdinando Fioretto, and Pascal Van Hentenryck. Fairness increases adversarial vulnerability. *arXiv preprint arXiv:2211.11835*, 2022.

Khang Tran, Ferdinando Fioretto, Issa Khalil, My T Thai, and NhatHai Phan. Fairdp: Certified fairness with differential privacy. *arXiv preprint arXiv:2305.16474*, 2023.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SyxAb30cY7.

Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin'ichi Satoh. Embedding watermarks into deep neural networks. In *ACM on International Conference on Multimedia Retrieval*, pp. 269–277, 2017. URL https://doi.org/10.1145/3078971.3078974.

Daniël Vos, Jelle Vos, Tianyu Li, Zekeriya Erkin, and Sicco Verwer. Differentially-private decision trees with probabilistic robustness to data poisoning. *arXiv preprint arXiv:2305.15394*, 2023.

Sandra Wachter, Brent Daniel Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Cybersecurity*, 2017. URL https://api.semanticscholar.org/CorpusID:3995299.

Zeming Wei, Yifei Wang, Yiwen Guo, and Yisen Wang. Cfa: Class-wise calibrated fair adversarial training. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8193–8201, 2023.

Emily Wenger, Shawn Shan, Haitao Zheng, and Ben Y Zhao. Sok: Anti-facial recognition technology. In *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 864–881. IEEE, 2023.

Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. In *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=4cEapqXfP30.

Jiapeng Wu, Atiyeh Ashari Ghomi, David Glukhov, Jesse C Cresswell, Franziska Boenisch, and Nicolas Papernot. Augment then smooth: Reconciling differential privacy with certified robustness. *arXiv preprint arXiv:2306.08656*, 2023.

Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *ArXiv*, abs/1802.06739, 2018. URL https://api.semanticscholar.org/CorpusID:3357865.

Shiji Xin, Yifei Wang, Jingtong Su, and Yisen Wang. On the connection between invariant learning and adversarial training for out-of-distribution generalization. In *AAAI Conference on Artificial Intelligence*, volume 37, pp. 10519–10527, 2023.

Chang Xu, Jun Wang, Francisco Guzmán, Benjamin Rubinstein, and Trevor Cohn. Mitigating data poisoning in text classification with differential privacy. In *Findings of the Association for Computational Linguistics*, pp. 4348–4356, November 2021a. URL https://aclanthology.org/2021.findings-emnlp.369.

Depeng Xu, Shuhan Yuan, and Xintao Wu. Achieving differential privacy and fairness in logistic regression. In *World Wide Web Conference*, pp. 594–599, 2019a. URL https://doi.org/10.1145/3308560.3317584.

Depeng Xu, Wei Du, and Xintao Wu. Removing disparate impact of differentially private stochastic gradient descent on model accuracy. *arXiv preprint arXiv:2003.03699*, 2020.

Han Xu, Xiaorui Liu, Yaxin Li, Anil Jain, and Jiliang Tang. To be robust or to be fair: Towards fairness in adversarial training. In *International Conference on Machine Learning*, volume 139, pp. 11492–11501, 18–24 Jul 2021b. URL https://proceedings.mlr.press/v139/xu21b.html.

Yilun Xu, Peng Cao, Yuqing Kong, and Yizhou Wang. L_dmi: An information-theoretic noise-robust loss function. *arXiv preprint arXiv:1909.03388*, 2019b.

Mohammad Yaghini, Patty Liu, Franziska Boenisch, and Nicolas Papernot. Learning with impartiality to walk on the pareto frontier of fairness, privacy, and utility. *arXiv preprint arXiv:2302.09183*, 2023.

Mohammad Yaghini, Patty Liu, Franziska Boenisch, and Nicolas Papernot. Regulation games for trustworthy machine learning. *arXiv preprint arXiv:2402.03540*, 2024.

Fan Yang, Qizhang Feng, Kaixiong Zhou, Jiahao Chen, and Xia Hu. Differentially private counterfactuals via functional mechanism. *arXiv preprint arXiv:2208.02878*, 2022.

Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Russ R Salakhutdinov, and Kamalika Chaudhuri. A closer look at accuracy vs. robustness. In *Advances in Neural Information Processing Systems*, pp. 8588–8601, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/61d77652c97ef636343742fc3dcf3ba9-Paper.pdf.

Jiayuan Ye, Zhenyu Zhu, Fanghui Liu, Reza Shokri, and Volkan Cevher. Initialization matters: Privacy-utility analysis of overparameterized neural networks. In *Advances in Neural Information Processing Systems*, volume 36, pp. 5419–5446, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1165af8b913fb836c6280b42d6e0084f-Paper-Conference.pdf.

Ashkan Yousefpour et al. Opacus: User-friendly differential privacy library in pytorch. *arXiv preprint arXiv:2109.12298*, 2021.

S. Yun, D. Han, S. Chun, S. Oh, Y. Yoo, and J. Choe. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6022–6031, 2019. URL https://doi.ieeecomputersociety.org/10.1109/ICCV.2019.00612.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. Fairness constraints: A flexible approach for fair classification. *Journal of Machine Learning Research*, 20(75):1–42, 2019. URL http://jmlr.org/papers/v20/18-262.html.

Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International Conference on Machine Learning*, volume 28, pp. 325–333, 17–19 Jun 2013. URL https://proceedings.mlr.press/v28/zemel13.html.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *AAAI/ACM Conference on AI, Ethics, and Society*, pp. 335–340, 2018a.

Hongjing Zhang and Ian Davidson. Towards fair deep anomaly detection. In *ACM conference on fairness, accountability, and transparency*, pp. 138–148, 2021.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, volume 97, pp. 7472–7482, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/zhang19p.html.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018b. URL https://openreview.net/forum?id=r1Ddp1-Rb.

Jialong Zhang, Zhongshu Gu, Jiyong Jang, Hui Wu, Marc Ph. Stoecklin, Heqing Huang, and Ian Molloy. Protecting intellectual property of deep neural networks with watermarking. In *Asia Conference on Computer and Communications Security*, pp. 159–172, 2018c. URL https://doi.org/10.1145/3196494.3196550.

Tao Zhang, Tianqing Zhu, Kun Gao, Wanlei Zhou, and S Yu Philip. Balancing learning model privacy, fairness, and accuracy with early stopping criteria. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

Yuan Zhang and Zhiqi Bu. Differentially private optimizers can learn adversarially robust models. *arXiv preprint arXiv:2211.08942*, 2022.

Runkai Zheng, Rongjun Tang, Jianze Li, and Li Liu. Data-free backdoor removal based on channel lipschitzness. In *European Conference on Computer Vision*, pp. 175–191. Springer, 2022a.

Runkai Zheng, Rongjun Tang, Jianze Li, and Li Liu. Pre-activation distributions expose backdoor neurons. In *Advances in Neural Information Processing Systems*, 2022b. URL https://openreview.net/forum?id=wwW-1k1ljIg.

Tianhang Zheng and Baochun Li. Differentially private dataset condensation. In *AISCC–Network and Distributed Systems Security*, 2023. URL https://openreview.net/forum?id=H8XpqEkbua_.

Yue Zheng, Si Wang, and Chip-Hong Chang. A dnn fingerprint for non-repudiable model ownership identification and piracy detection. *IEEE Transactions on Information Forensics and Security*, 17:2977–2989, 2022c.

Mingli Zhu, Shaokui Wei, Hongyuan Zha, and Baoyuan Wu. Neural polarizer: A lightweight and effective backdoor defense via purifying poisoned features. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=VFhN15Vlkj.

A Summary of Defenses and Notations

We summarize the different defenses and their impact on ϕ_u from §3.1 in Table 8.

B Formal Analysis of Def\Con

A defense $\mathtt{D}$ is defined as a tuple $D := (S, C, R, P)$, where:

- S is the stage where D is applied
- C is the type of changes
- R is the risk it uses
- P is the protection scope of D, and $\emptyset \notin P$.

Let $\mathcal{D}$ be the set of all such defense. The input set for Def\Con is defined as: $X := \{(\mathtt{D}_1, \mathtt{D}_2) | \mathtt{D}_1 \neq \mathtt{D}_2, \textit{ and } \mathtt{D}_1, \mathtt{D}_2 \in \mathcal{D}\}$. Def\Con can be defined as a function: $f : X \to \{0, 1\}$ where $f(\mathtt{D}_1, \mathtt{D}_2) = 0$ indicates conflict and $f(\mathtt{D}_1, \mathtt{D}_2) = 1$ indicates alignment. Given the above formal model, we will discuss consistency, soundness, and completeness.

Consistency means that Def\Con never produces contradictory outputs for the same input pair.

Claim 1. *There is no pair $(\mathtt{D}_1, \mathtt{D}_2)$ for which Def\Con simultaneously classifies both conflict and alignment, i.e. Def\Con is consistent.*

Proof. According to the definition of f, There are two rules or conflict cases:

Table 8: **Summary of defenses.** (Column ϕ_u indicates impact on utility: "$\vee$" (decrease), "$\sim$" (no effect), "$\wedge$" $\rightarrow$ (increase).)

Defense	ϕ_u	References
EvsnRob (Evasion Robustness)		
• EvsnRob.Pre (Data Augmentation)	$\wedge$	Yun et al. (2019); Zhang et al. (2018b); DeVries & Taylor (2017); Madry et al. (2018); Rebuffi et al. (2021)
• EvsnRob.In (Adversarial Training)	$\vee$	Zhang et al. (2019); Cohen et al. (2019); Lecuyer et al. (2019)
• EvsnRob.Post (Input Processing)	$\vee$	Nie et al. (2022); Song et al. (2018); Guo et al. (2018); Das et al. (2017)
	$\sim$	Grosse et al. (2017); Buckman et al. (2018)
PoisnRob (Outlier Robustness)		
• PoisnRob.Pre (Data Augmentation)	$\vee$	Borgnia et al. (2021); Qiu et al. (2021); Cretu et al. (2008); Paudice et al. (2018; 2019); Jia et al. (2021b; 2019)
• PoisnRob.In (Fine-tuning)	$\sim$	Diakonikolas et al. (2019); Xu et al. (2019b); Liu & Guo (2020); Patrini et al. (2017); Zhu et al. (2023); Liu et al. (2018); Wu & Wang (2021); Li et al. (2017)
• PoisnRob.Post (Pruning)	$\vee$	Zheng et al. (2022b;a); Li et al. (2023c)
MdlWM (Watermarking-M)		
• MdlWM.Pre (Backdoors)	$\sim$	Adi et al. (2018); Zhang et al. (2018c); Jia et al. (2021a); Uchida et al. (2017)
• MdlWM.In (Optimization)	$\vee$	Bansal et al. (2022)
• MdlWM.Post (API-based)	$\sim$	Szyller et al. (2021)
Fngrprnt (Fingerprinting)		
• Fngrprnt.Post (Fingerprints)	$\sim$	Cao et al. (2021); Peng et al. (2022); Lukas et al. (2021); Zheng et al. (2022c); Maini et al. (2021)
DtWM (Watermarking-D)		
• DtWM.Pre (Backdoors)	$\sim$	Tekgul & Asokan (2022); Sablayrolles et al. (2020); Liu et al. (2022a)
DiffPriv (Differential Privacy)		
• DiffPriv.Pre (Private Data)	$\vee$	Xie et al. (2018); Torkzadehmahani et al. (2019)
• DiffPriv.In (DPSGD)	$\vee$	Abadi et al. (2016); Papernot et al. (2017)
GpFair (Group Fairness)		
• GpFair.Pre (Fair Data)	$\vee$	Kamiran & Calders (2011); Calmon et al. (2017); Zemel et al. (2013); Feldman et al. (2015)
• GpFair.In (Regularization)	$\vee$	Celis et al. (2019); Kearns et al. (2018; 2019); Agarwal et al. (2019; 2018); Zhang et al. (2018a); Kamishima et al. (2012)
• GpFair.Post (Calibration)	$\vee$	Pleiss et al. (2017); Hardt et al. (2016); Kamiran et al. (2012); Geyik et al. (2019)
Expl (Explanations)		
• Expl.Post (Attributions)	$\sim$	Ismail et al. (2021); Smilkov et al. (2017); Sundararajan et al. (2017); Koh & Liang (2017); Wachter et al. (2017); Selvaraju et al. (2017); Kim et al. (2018)

- (c.1) If $S_1 = S_2$ and $C_1 = Global$ and $C_2 = Global$, then $f(\mathtt{D}_1, \mathtt{D}_2) = 0$.
- (c.2) If $S_1 \neq S_2$ and $R_1 \in P_2$, then $f(\mathtt{D}_1, \mathtt{D}_2) = 0$.

For alignment case: (a.1) If none of the conflict conditions hold, then $f(\mathtt{D}_1, \mathtt{D}_2) = 1$.

These conditions partition the input space X into disjoint sets: conflict or alignment, with no overlap. Assume that there exists X_i such that $f(X_i) = 0$ and $f(X_i) = 1$. Since $f(X_i) = 1$, according to (a.1), (c.1) and (c.2) do not hold, leading to a contradiction. Therefore, Def\Con never produces contradictory classifications for the same input and is consistent. □

Soundness ensures that if Def\Con predicts a combination as effective (aligned), it should indeed be effective in practice.

This assumption is supported by our empirical evaluation, where Def\Con achieves:

- 90% accuracy on previously studied defense combinations,
- 81% accuracy on novel, unexplored combinations.

The low rates of false positives and false negatives in these evaluations suggest that Def\Con can reliably predict combination effectiveness, providing practical evidence for its soundness. A perfectly sound technique would rely on comprehensive empirical evaluation which we want to avoid by proposing an easy-to-use (approximate) technique, Def\Con. Therefore, the soundness of Def\Con is conditional on the assumption that these rules perfectly represent all conflict and alignment scenarios: (c.1) (c.2) and (a.1). However, since real-world defense interactions can be complex, involving subtle dependencies and emergent behaviors,

DEF\CON may not fully captured reasons for conflict or alignment. Hence, DEF\CON is not perfectly sound and is likely to incur some errors.

Completeness suggests that DEF\CON can classify *every possible defense pair* correctly as either conflict or alignment. Similar to soundness, completeness is conditional on the assumption that the classification rules fully capture all conflict and alignment cases. As a partial theoretical guarantee, we can prove that DEF\CON can classify every defense pair in its input domain, i.e., it produces a classification for every pair without leaving any case unclassified. Assuming that the input set X includes all possible distinct defense pairs, DEF\CON partially satisfies completeness by design. Formally,

Claim 2. *For every pair $(\mathtt{D}_1, \mathtt{D}_2) \in X$, DEF\CON's function f produces a classification $f(\mathtt{D}_1, \mathtt{D}_2) \in \{0, 1\}$ that identifies whether the defenses conflict or align, i.e., no conflict or alignment case is left unclassified. Therefore, if X includes all distinct combinations of defenses, DEF\CON is partially complete.*

Proof. By construction, DEF\CON 's classification rules partition the input space X exhaustively and exclusively by (c.1) (c.2) and (a.1). These rules cover all possible defense pairs in X without overlap or ambiguity, ensuring that every pair is assigned a unique classification, DEF\CON is partially complete on X. □

The assumption that X includes all possible defense combinations is supported by the fact that all surveyed defense methods can be represented within the DEF\CON framework.

C Formal Analysis for Multi-way Combination Algorithm

The algorithm F is for multi-way combinations and iterates through all permutations of defenses, and uses DEF\CON to check for pairwise conflicts:

(m.1) For defenses $\mathtt{D}_i \in \mathcal{D}$, $i = 1, ..., n$, $F(\mathtt{D}_1, ..., \mathtt{D}_n) = 0$ iff $\exists \mathtt{D}_i, \mathtt{D}_j \in \mathtt{D},\ f(\mathtt{D}_i, \mathtt{D}_j) = 0$.

Consistency. According to the consistency of f, assume that there exists $\mathtt{D}_i, i = 1, ..., n$, s.t. $F(\mathtt{D}_1, ..., \mathtt{D}_n) = 0$ and $F(\mathtt{D}_1, ..., \mathtt{D}_n) = 1$. Since $F(\mathtt{D}_1, ..., \mathtt{D}_n) = 1$, $\forall \mathtt{D}_i, \mathtt{D}_j$, we have $f(\mathtt{D_i}, \mathtt{D}_j) = 1$, leading to a contradiction. Therefore, F is also consistent.

Soundness and Completeness. Under the assumption that f is sound and complete, and considering that we take all possible combinations of defenses (as there is no limit on the input defense set of F), the soundness and completeness of F is conditional on the assumption that (m.1) fully captures all conflict and alignment cases.

Soundness. Since f has some errors and is not perfectly sound, multi-way combinations can indeed introduce conflicts, which will result in false positives or negatives. Hence, the soundness of of multi-way DEF\CON requires comprehensive evaluation similar to pairwise evaluation. In our evaluation, we only want to show that effective multi-way combinations—previously unexplored—are possible. A comprehensive evaluation to check for soundness is left as future work.